

HÉLICES EN LA NATURALEZA

AUTORES

Tasio Ruano Arregui

GRADO EN INGENIERÍA MATEMÁTICA, FACULTAD DE CIENCIAS MATEMÁTICAS,
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin grado

25/02/2015

Directores:

Ana Carpio

Autorización de Difusión

TASIO RUANO

25/02/2015

El/la abajo firmante, matriculado/a en el Grado en Ingeniería Matemática de la Facultad de Ciencias Matemáticas, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: “HÉLICES EN LA NATURALEZA”, realizado durante el curso académico 2014-2015 bajo la dirección de Ana Carpio en el Departamento de Matemática Aplicada, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Resumen en castellano

El objetivo de este trabajo es estudiar formulaciones matemáticas sencillas que permitan simular la dinámica de filamentos elásticos. Estos filamentos se considerarán inextensibles para simplificar la formulación y la resolución del problema. Nos centraremos en teorías que parametrizan los filamentos mediante curvas y sistemas de referencia adaptados, y a partir de los cuales seremos capaces de plantear y medir las magnitudes físicas que intervienen en el comportamiento del filamento y posteriormente a partir de éstas hallar una distribución de las energías existentes en el filamento, que serán las que determinen el movimiento de la misma.

A continuación exploraremos una versión discreta. En ella extrapolaremos todo lo obtenido en la teoría continua desarrollada anteriormente, transformando la curva continua en un conjunto de puntos y definiendo sistemas de referencia adaptados en cada uno de los segmentos que los unen. Una vez asentadas en nuestra teoría discreta todas las magnitudes usadas en la continua plantearemos las ecuaciones del movimiento.

Consideraremos la transformación de estas teorías en algoritmos programables y desarrollaremos un código. Se trabajará con modelos en los que las ondas de torsión en la cuerda se propagan instantáneamente en relación con las ondas de flexión, con lo que se simplificará el problema no calculando su evolución temporal y se ahorrará un coste computacional muy valioso. Esto último junto con el método de proyección múltiple reducirá considerablemente la complejidad del problema, permitiendo simular situaciones más complejas en un tiempo razonable.

Dada una ecuación diferencial rígida, los métodos de proyección múltiple son una alternativa a los métodos de integración implícitos, y consisten en reformular la componente rígida del problema como una restricción. Combinaremos un método Verlet simpléctico de orden dos para resolver las ecuaciones del movimiento con la técnica de proyección múltiple, que se aplicará a cada instante de tiempo del problema para forzar la inextensibilidad.

Por último, se mostrará un pequeño ejemplo de una simulación junto con su código.

Palabras clave

Elasticidad, filamentos, simulación

Resumen en inglés

The aim of this work is to study simple mathematical formulations which allow us to simulate the dynamics of elastic threads. These threads will be considered to be inextensible to simplify both the formulation and solution of the problem. We will focus on theories which parameterize threads by curves and adapted frames, from which we will be able to formulate and measure the physical values that participate in the thread's behavior. Afterwards we will find an energy distribution for the thread from these values, distribution that will establish the thread's motion.

Next we will explore the discrete version. We will extrapolate the results produced by the continuous theory developed beforehand, transforming the continuous curve into a set of points and defining adapted frames on the segments that connect them. Once all magnitudes used in the continuous theory are stated we will formulate the equations of motion.

We will transform these equations into programmable algorithms and develop a programming code. We will work with models in which twist waves propagate instantaneously when compared with bending waves. This simplifies the problem reducing the computational cost. This fact together with the manifold projection method will greatly reduce problem's complexity, and will allow us simulate more complex situations in a reasonable amount of time.

Given a stiff differential equation, manifold projection methods are an alternative to implicit integration methods, and they consist in reformulating the problem's stiff component as a restriction. Our particular method will combine a symplectic Verlet method of second order with the manifold projection scheme, which we will use in every discrete step of time to enforce the inextensibility hypothesis.

Finally, we will show a brief example of a simulation along with the programming code.

Keywords

Elasticity, threads, simulation

Índice de contenidos

Autorización de Difusión	ii
Resumen en castellano	iii
Resumen en inglés	iv
Índice de contenidos	1
Capítulo 1 - Teorías continua y discreta	3
Teoría continua	3
Representación de la cuerda.....	3
Energías elásticas en la cuerda.....	4
Sistema Bishop y transporte paralelo.....	6
Teoría discreta de la cuerda	10
Representación de la cuerda.....	10
Holonomía discreta	12
Energía discreta de flexion.....	15
Transporte paralelo y sistemas Bishop discretos	16
Energía de torsión	18
Condiciones de contorno e hipótesis previas	19
Cálculo de ángulos y actualización de la cuerda.....	20
Ecuaciones del movimiento	21
Capítulo 2 - Simulación de la cuerda elástica discreta	24
Introducción de los datos iniciales y en el equilibrio.....	25
Construcción de los datos en el equilibrio	27
Construcción de los datos iniciales	30
Cálculo de las energías iniciales	33
Proyección múltiple	34
Evolución temporal	36
Capítulo 3 – Código y ejemplos.....	37
Conclusión	49
References Or Bibliography	51

Capítulo 1 - Teorías continua y discreta

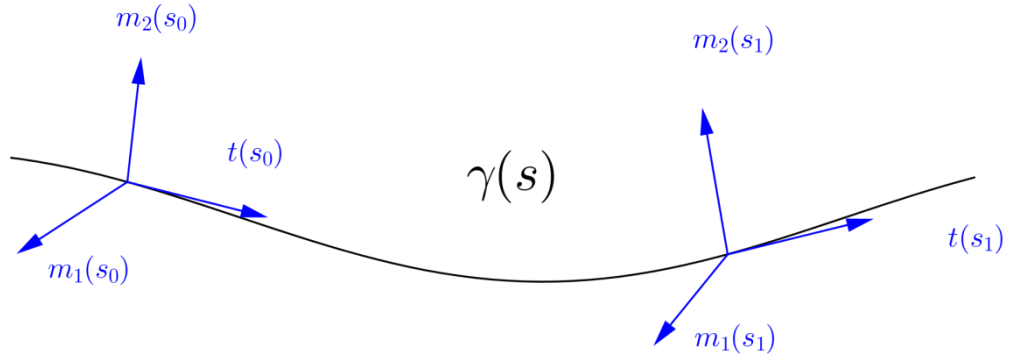
Este capítulo establecerá una base teórica de la modelización de cuerdas y hélices para sustentar el desarrollo posterior del algoritmo. Primero explicaremos la teoría continua mediante la que modelizamos la cuerda, y luego a partir de ésta nos centraremos en la teoría discreta que será la que nos permita generar el algoritmo de simulación. Las cuerdas vendrán dadas por curvas paramétricas y se aplicarán conceptos de geometría diferencial para medir los distintos tipos de energías que soportan, y posteriormente se extrapolará a una teoría discreta convenientemente escogida para que el algoritmo después empleado sea eficiente.

Conviene recalcar que este capítulo es sólo una introducción teórica para la posterior aplicación algorítmica y no profundizaremos demasiado en ella, ya que por sí sola ocuparía varios trabajos como éste. De cualquier modo, servirá para comprender qué subyace detrás del algoritmo y para profundizar en él, o para hacer posibles modificaciones.

Teoría continua

Representación de la cuerda

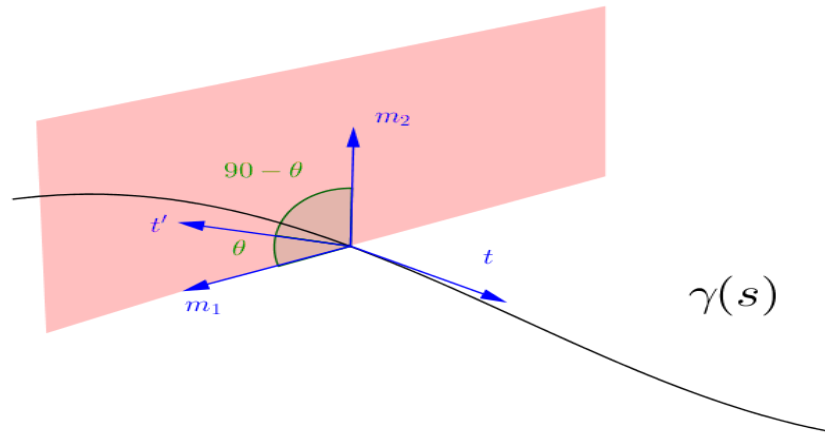
Consideraremos una cuerda cualquiera con una distribución de masas $\mathbf{m}(\mathbf{x})$. Consideraremos, para facilitar todo el cálculo posterior, que se trata de una cuerda sin grosor alguno. Esto se puede hacer reduciendo la cuerda con volumen a la curva que ocuparía el centro de la misma. Con esto, tenemos que la posición de la cuerda se puede representar mediante una curva paramétrica $\mathbf{y}(s)$, parametrizada por la longitud de arco. Ahora, para cada punto de la curva, podemos asignar un sistema ortonormal de 3D $\{\mathbf{t}(s), \mathbf{m}_1(s), \mathbf{m}_2(s)\}$ al que llamaremos sistema material que nos permita medir interesantes propiedades físicas de la cuerda, tales como su curvatura o su torsión. Así, tomamos $\mathbf{t}(s) = \mathbf{y}'(s)$, que medirá la curvatura de la cuerda, mientras que los otros dos vectores estarán relacionados con la torsión. Por tanto, cualquier punto de la curva quedará físicamente descrito por $\Gamma = \{\mathbf{y}(s); \mathbf{t}(s), \mathbf{m}_1(s), \mathbf{m}_2(s)\}$, tal y como muestra la siguiente figura:



Energías elásticas en la cuerda

En la cuerda existirán dos tipos de energías relacionadas con la elasticidad de la cuerda: la energía de doblado y la energía de torsión. La teoría de Kirchhoff de cuerdas elásticas asigna a la cuerda una energía elástica $E(\Gamma)$ dada por tres funciones escalares que miden las tensiones de ésta a partir de los cambios en el sistema material. Éstas son:

$$\omega_1(s) = \mathbf{t}'(s) \mathbf{m}_1(s), \quad \omega_2(s) = \mathbf{t}'(s) \mathbf{m}_2(s), \quad m(s) = \mathbf{m}_1'(s) \mathbf{m}_2(s)$$



Nótese que como $\mathbf{t}'(s) = \kappa(s)$, los dos primeros términos miden el doblado o flexión de la cuerda (o del sistema material), mientras que el último mide la torsión alrededor de la tangente.

La energía total de la cuerda puede ser expresada a partir de la suma de la energía de flexión y la de torsión:

$$E(\Gamma) = E_{\text{flexión}}(\Gamma) + E_{\text{torsión}}(\Gamma)$$

Asumiremos que nuestra cuerda es inextensible por lo que no incluiremos una energía a causa del estiramiento de la cuerda. Esto se puede hacer a partir de restricciones auxiliares, con lo que disminuiríamos el coste computacional significativamente con relación a una cuerda extensible.

Cuando la configuración en reposo de la cuerda es recta e isotrópica, es decir, ésta no favorece giros en ningún sentido en particular, la energía de flexión viene dada por

$$E_{\text{flexión}}(\Gamma) = \frac{1}{2} \int \alpha \omega^2(s) ds$$

Con $\omega = (\omega_1, \omega_2)^T$ la curvatura material y α el módulo o resistencia de flexión de la cuerda. Nótese que el vector ω representa la curvatura de la curva en función del sistema material.

Podemos reexpresar el integrando, de manera que

$$\begin{aligned} \omega^2 &= \omega_1^2 + \omega_2^2 = (\mathbf{t}' \mathbf{m}_1)^2 + (\mathbf{t}' \mathbf{m}_2)^2 = \|\mathbf{t}'\|^2 \|\mathbf{m}_1\|^2 \cos^2(\theta) + \|\mathbf{t}'\|^2 \|\mathbf{m}_2\|^2 \cos^2(90 - \theta) \\ &= \|\mathbf{t}'\|^2 (\|\mathbf{m}_1\|^2 \cos^2(\theta) + \|\mathbf{m}_2\|^2 \sin^2 \theta) = \|\mathbf{t}'\|^2 = \kappa^2 \end{aligned}$$

Ya que la derivada de la tangente pertenece al plano generado por los vectores \mathbf{m}_1 y \mathbf{m}_2 y al tratarse de un sistema ortonormal $\|\mathbf{m}_1\|^2 = \|\mathbf{m}_2\|^2 = 1$. Podemos escribir entonces

$$E_{\text{flexión}}(\Gamma) = \frac{1}{2} \int \alpha \omega^2(s) ds = \frac{1}{2} \int \alpha \kappa^2(s) ds$$

Generalizamos este resultado para cuerdas no-isotrópicas reemplazando el producto escalar usual por una forma cuadrática definida positiva \bar{B} , de manera que $\omega^T(s) \bar{B} \omega(s)$ es la densidad de energía de flexión. Podemos generalizar este resultado a una cuerda cuya configuración en reposo sea curva sin más que reemplazar $\omega(s)$ por $(\omega(s) - \bar{\omega}(s))$, donde $\bar{\omega}(s)$ es la curvatura de la cuerda en su configuración de reposo.

Recordemos que $m(s) = \mathbf{m}_1'(s)\mathbf{m}_2(s)$. La energía de torsión de la cuerda vendrá dada por

$$E_{\text{torsión}}(\Gamma) = \frac{1}{2} \int \beta m^2(s) ds$$

Recordemos que la torsión de una curva viene dada por

$$\tau(s) = -\mathbf{n}(s)\mathbf{b}'(s)$$

Donde $\mathbf{n}(s)$ es el vector normal a la curva y $\mathbf{b}'(s)$ la derivada del binormal. Estos dos son parte del sistema de Frenet, que tiene en común con nuestro sistema material a la tangente. Por tanto, podemos expresar uno en función del otro como una rotación de ángulo φ alrededor de la tangente, de manera que

$$\mathbf{n}(s) = \mathbf{m}_1(s) \cos \varphi(s) + \mathbf{m}_2(s) \sin \varphi(s),$$

$$\mathbf{b}(s) = -\mathbf{m}_1(s) \sin \varphi(s) + \mathbf{m}_2(s) \cos \varphi(s)$$

$$\begin{aligned} -\mathbf{b}'(s) &= \mathbf{m}_1(s)\varphi'(s) \cos \varphi(s) + \mathbf{m}_1'(s) \sin \varphi(s) + \mathbf{m}_2(s)\varphi'(s) \sin \varphi(s) - \\ &\quad - \mathbf{m}_2'(s) \cos \varphi(s) \end{aligned}$$

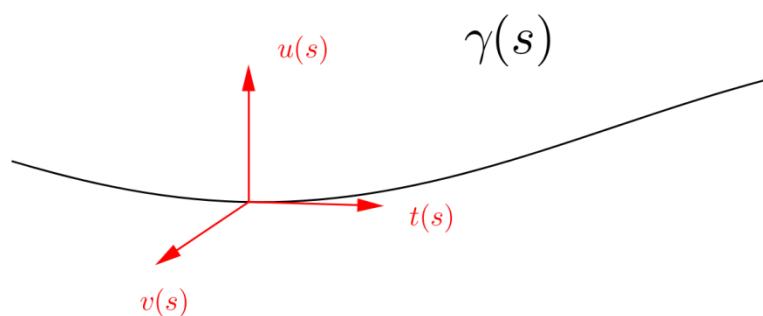
Y la torsión viene dada entonces en términos de nuestro sistema material por

$$\begin{aligned} \tau(s) &= -\mathbf{n}(s)\mathbf{b}'(s) = \\ &= \varphi'(s) \cos^2 \varphi(s) - (-m(s)) \cos^2 \varphi(s) + m(s) \sin^2 \varphi(s) + \\ &\quad + \varphi'(s) \sin^2 \varphi(s) = \varphi'(s) + m(s) \end{aligned}$$

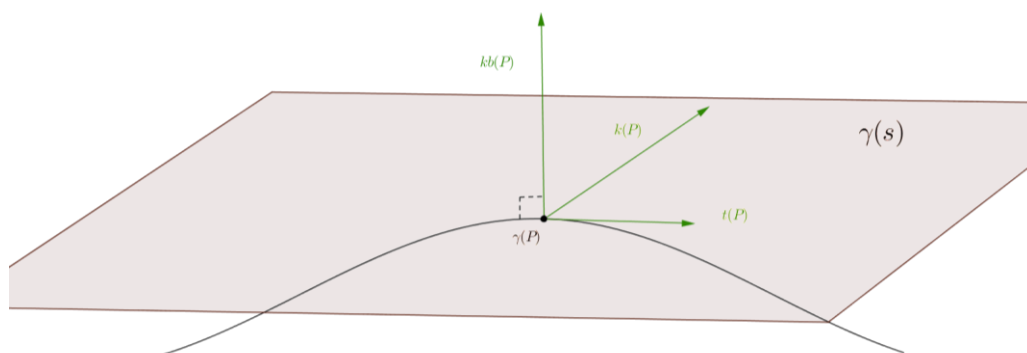
De este modo, vemos que la torsión geométrica y nuestro concepto de torsión están estrechamente relacionados.

Sistema Bishop y transporte paralelo

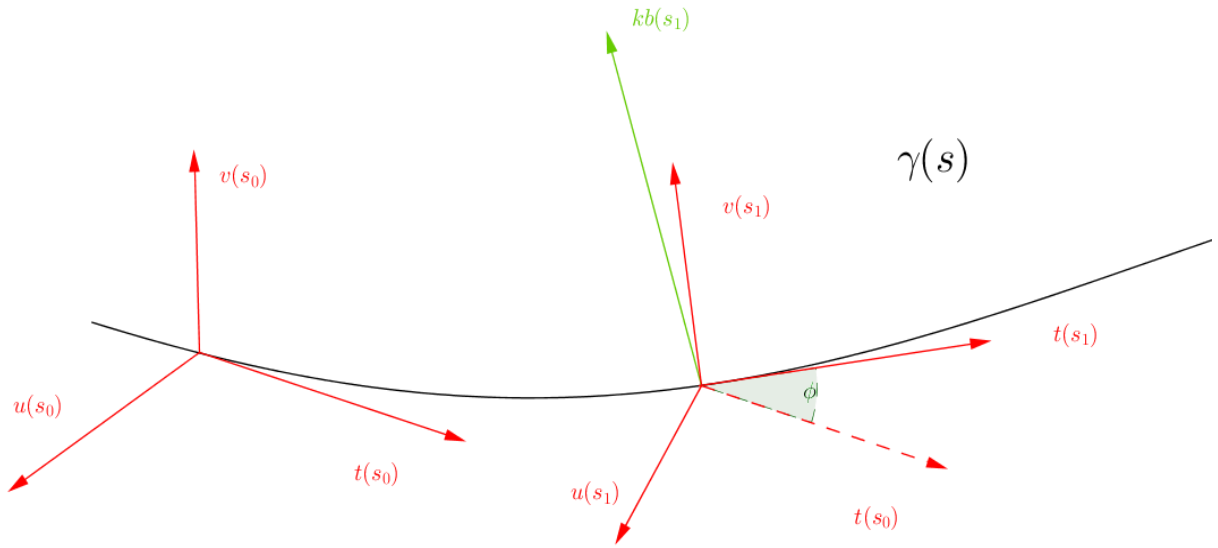
Dada ahora una posición fija, nos planteamos el asignarle a cada punto de la cuerda un sistema de referencia ortonormal que sea lo más natural y relajado posible, en el que las tensiones y otro tipo de energías sean mínimas. A este nuevo sistema lo llamaremos Bishop $\{\mathbf{t}(s), \mathbf{u}(s), \mathbf{v}(s)\}$, y cumplirá que su torsión será cero a lo largo de la cuerda, o lo que es lo mismo, $\mathbf{u}'(s)\mathbf{v}(s) = -\mathbf{u}(s)\mathbf{v}'(s) = 0$.



Este sistema se puede asignar sin más que fijar el sistema Bishop al inicio de la cuerda, y a partir de ahí transportarlo con lo que llamaremos transporte paralelo. Éste viene dado en términos del vector de Darboux o curvatura binormal, dado por $k\mathbf{b} = \mathbf{t} \times \mathbf{k}$ (es decir, el binormal con la curvatura por módulo), pero aplicado a nuestro trabajo usaremos una definición más sencilla.



El transporte paralelo consiste en una rotación alrededor del vector binormal de la cuerda (o vector de Darboux, ya que tienen misma dirección y sentido), de manera que dado el primer sistema Bishop y calculando la tangente en cualquier otro punto de la cuerda podemos hallar el ángulo de giro ϕ entre ambas tangentes y calcular la rotación $\mathbf{P}(\phi)$ alrededor del binormal para transformar una tangente en otra, y así conseguir el sistema Bishop en el nuevo punto de la cuerda.



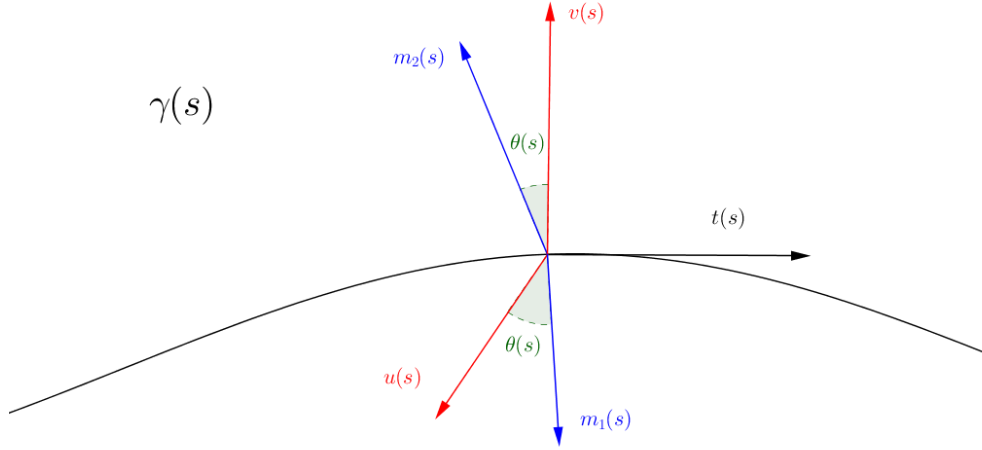
Escribamos ahora las dos componentes del sistema material antes descrito en función de las componentes del sistema Bishop. Teniendo en cuenta que la tangente a la curva es un vector común a ambos y que ambos sistemas son ortonormales, necesariamente el sistema material es el resultado de una rotación alrededor de la tangente del sistema Bishop para un cierto ángulo dado por una función $\theta(s)$, de manera que

$$\mathbf{m}_1(s) = \mathbf{u}(s) \cos \theta(s) + \mathbf{v}(s) \sin \theta(s), \quad \mathbf{m}_2(s) = -\mathbf{u}(s) \sin \theta(s) + \mathbf{v}(s) \cos \theta(s)$$

Y de manera análoga:

$$\mathbf{u}(s) = \mathbf{m}_1(s) \cos \theta(s) - \mathbf{m}_2(s) \sin \theta(s), \quad \mathbf{v}(s) = \mathbf{m}_1(s) \sin \theta(s) + \mathbf{m}_2(s) \cos \theta(s)$$

Podemos observar la relación entre ambos en la siguiente figura:



Queremos ahora observar la relación existente entre $m(s)$ y $\theta(s)$. De hallar alguna, simplificaríamos en gran medida los cálculos, ya que la energía de torsión sólo estaría relacionada con el ángulo existente entre el sistema Bishop y el natural. Para ello, usaremos que $\mathbf{u}'(s)\mathbf{v}(s) = 0$

$$\mathbf{u}'(s) = -\mathbf{m}_1(s)\theta'(s)\sin\theta(s) + \mathbf{m}_1'(s)\cos\theta(s) - \mathbf{m}_2(s)\theta'(s)\cos\theta(s) - \mathbf{m}_2'(s)\sin\theta(s)$$

$$0 = \mathbf{u}'(s)\mathbf{v}(s) =$$

$$\begin{aligned} &= -\mathbf{m}_1(s)\mathbf{m}_2(s)\theta'(s)\sin\theta(s)\cos\theta(s) + \mathbf{m}_1'(s)\mathbf{m}_2(s)\cos^2\theta(s) - \\ &- \theta'(s)\cos^2\theta(s)\mathbf{m}_2(s)\mathbf{m}_2(s) - \mathbf{m}_2(s)\mathbf{m}_2'(s)\sin\theta(s)\cos\theta(s) - \\ &- \mathbf{m}_1(s)\mathbf{m}_1(s)\theta'(s)\sin^2\theta(s) + \mathbf{m}_1(s)\mathbf{m}_1'(s)\sin\theta(s)\cos\theta(s) - \\ &- \mathbf{m}_1(s)\mathbf{m}_2(s)\theta'(s)\sin\theta(s)\cos\theta(s) - \mathbf{m}_1(s)\mathbf{m}_2'(s)\sin^2\theta(s) \end{aligned}$$

Teniendo en cuenta que $\mathbf{m}_1(s)\mathbf{m}_1'(s) = \mathbf{m}_1(s)\mathbf{m}_2(s) = 0$, que $\cos^2\theta(s) + \sin^2\theta(s) = 1$ y que $m(s) = \mathbf{m}_1'(s)\mathbf{m}_2(s) = -\mathbf{m}_1(s)\mathbf{m}_2'(s)$, operamos y llegamos a que

$$0 = m(s) - \theta'(s) \Leftrightarrow m(s) = \theta'(s)$$

De este modo, podemos reexpresar el término de la energía asociado a la torsión como

$$E_{\text{torsión}}(\Gamma) = \frac{1}{2} \int \beta \theta'^2(s) ds$$

Teoría discreta de la cuerda

Representación de la cuerda

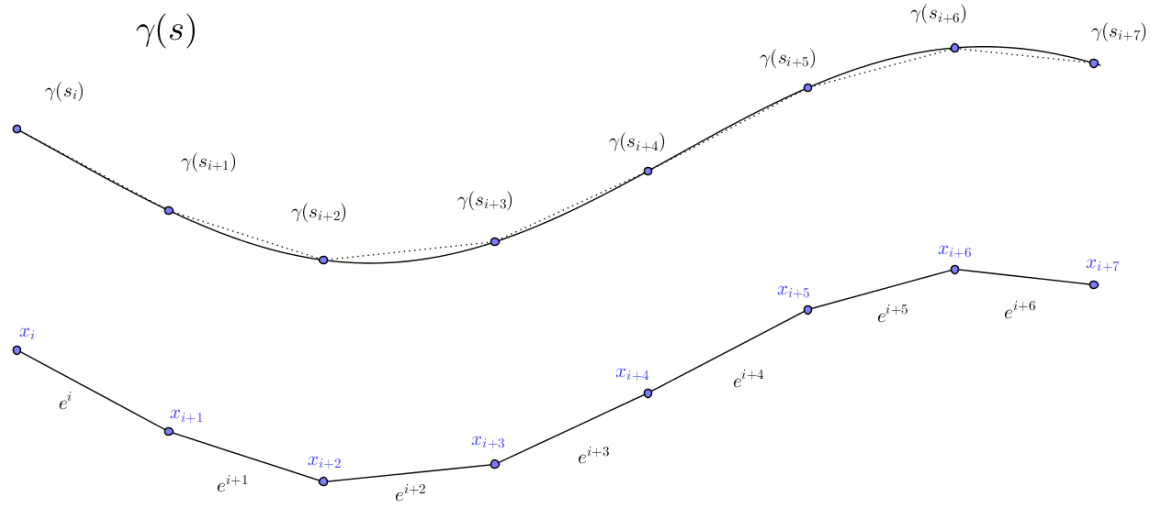
Ya tenemos las herramientas necesarias para determinar el comportamiento de una cuerda en el caso continuo. El problema consiste ahora en extrapolar los resultados de la teoría continua a la discreta, para que un ordenador pueda simular y reproducir el proceso.

Sea ahora la misma cuerda discretizada, es decir, una colección de puntos en el espacio tridimensional. Ahora, calcular las derivadas de la curva como en el caso continuo nos resulta imposible, ya que no disponemos de función alguna, con lo que los conceptos de tangente y demás vectores diferenciables tendrán que ser ligeramente modificados. Nos basaremos en la idea intuitiva de la derivada, que es la idea del incremento. Llamemos al conjunto de puntos de la cuerda discretizada X , y a cada uno de los n puntos o vértices lo llamaremos $x_i, i \in \{1, \dots, n\}$.

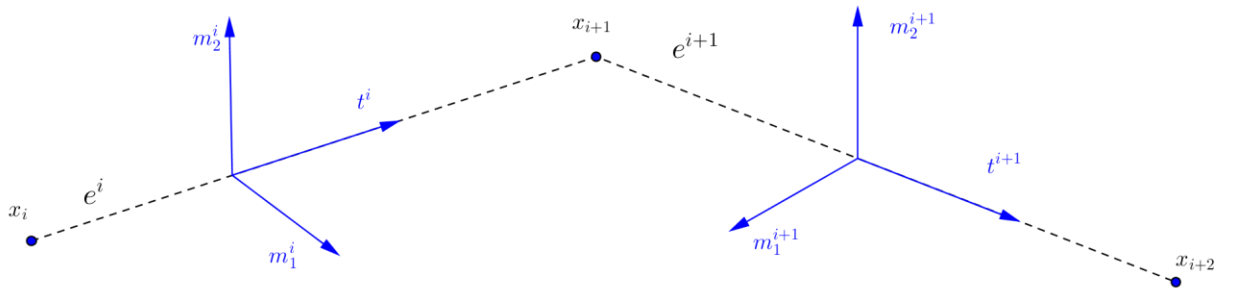
Sean $e^i, i \in \{1, \dots, n-1\}$ cada uno de los $(n-1)$ puentes entre los vértices, de manera que

$$e^i = x_{i+1} - x_i.$$

La siguiente figura muestra la manera mediante la que dada una curva continua la discretizamos:



Igual que en el caso continuo, nos planteamos el asignar a esta cuerda una versión discreta del sistema material. Para ello, en vez de ser funciones continuas, tendremos un conjunto de $(n-1)$ sistemas materiales, uno para cada uno de los puentes de la cuerda discreta. Así, denotamos $\mathbf{M}^i = \{\mathbf{t}^i, \mathbf{m}_1^i, \mathbf{m}_2^i\}$ al sistema material relativo al puente \mathbf{e}^i . Tomaremos $\mathbf{t}^i = \mathbf{e}^i / \|\mathbf{e}^i\|$, que por construcción es unitario como en el caso continuo, y es la mejor aproximación que podemos conseguir a un vector tangente.



Al no ser suave nuestro conjunto de puntos, los vectores tangentes estarán definidos únicamente en los puentes de la cuerda, y no así en los vértices de la misma, donde es más

complicado hallar una aproximación del vector tangente . De este modo, restringimos la asignación de sistemas materiales (y más adelante, Bishops) a los puentes existentes entre los vértices, sin importar lo que pase exactamente en estos últimos.

Ya hemos extrapolado la base existente en la teoría continua. Ahora, queda ver cómo transformamos las expresiones de las energías al caso discreto. En el caso continuo disponíamos de un dominio de integración D . Para convertir esto al caso discreto, definimos $(n-2)$ dominios D_i que incluyen los dos puentes incidentes en un mismo vértice. Llamaremos también l_i a la longitud de dos puentes consecutivos, es decir $l_i = \|e^i\| + \|e^{i+1}\|$, y diremos que $\|D_i\| = l_i/2$.

Decimos cantidad integrada a una medida que asocia un número x_i a D_i , que se corresponde con una integral de la función sobre ese dominio. Para transformar esa cantidad en una puntual, la dividimos entre la longitud del dominio D_i .

Por ejemplo, si tenemos las funciones continuas $f(x) = 3 * x^3 + x$, $y(s) = s$ en un dominio determinado, y quisiera calcular

$$\int_1^2 f(y(s))ds$$

Podría aproximarla a una forma discreta dividiendo en intervalo (1,2) en 10 dominios de longitud 0.1, y reexpresarla como

$$\int_1^2 f(y(s))ds \sim \sum_{i=1}^{10} \left(3 \left(\frac{y_i}{0.1} \right)^3 + \frac{y_i}{0.1} \right) * 0.1$$

Donde y_i es la cantidad integrada que aproxima el valor de la función $y(s)$ en el intervalo.

Por último, para referirnos a un valor o vector en el equilibrio utilizaremos la misma notación que la descrita hasta ahora, sin más que añadir una \mathbf{R} a su nombre, como por ejemplo $e\mathbf{R}^i$ se refiere al equilibrio del puente e^i .

Holonomía discreta

La holonomía es un concepto de geometría diferencial usado para medir la falta o pérdida de información geométrica al aplicar el transporte paralelo sobre giros muy cerrados. Lo denotaremos como $\psi(s)$, y se puede probar que $\delta(\int \psi ds) = - \int k\mathbf{b}(\delta\mathbf{x})_s ds$, donde el subíndice s denota diferenciación respecto al parámetro s de la curva. Esto quiere decir que está

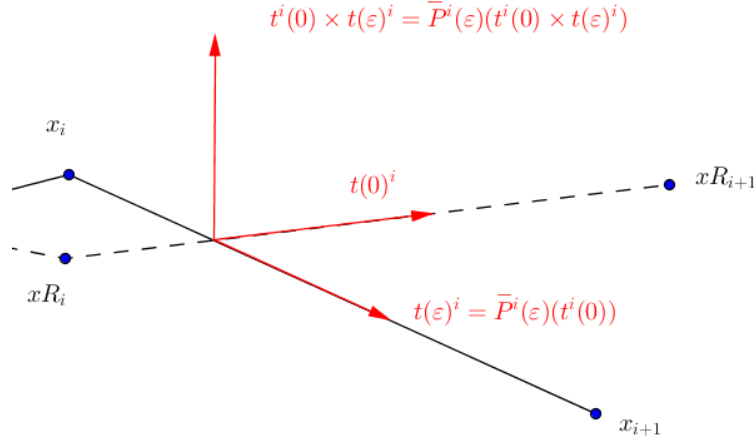
íntimamente relacionada con la curvatura de la cuerda. En nuestro caso discreto, sólo depende de la curva, y mide el ángulo cuando transportas paralelamente un sistema a lo largo de un giro cerrado de puentes. Consideramos dos puentes consecutivos e^{i-1}, e^i y sus tangentes asociadas t^{i-1}, t^i , a los cuales perturbaremos un parámetro ε , obteniendo nuevos vectores $e^{i-1}(\varepsilon), e^i(\varepsilon), t^{i-1}(\varepsilon), t^i(\varepsilon)$. Tenemos entonces una aplicación de transporte paralelo

$$P_i(\varepsilon) : t^{i-1}(\varepsilon) \rightarrow t^i(\varepsilon)$$

que transforma una tangente perturbada en otra. Ahora, consideramos el transporte paralelo de una tangente sin deformar a la tangente resultante perturbada, al que llamaremos

$$\tilde{P}^i(\varepsilon) : t^i(0) \rightarrow t^i(\varepsilon)$$

Esta rotación satisface $\tilde{P}^i(0) = Id$, $\tilde{P}^i(\varepsilon) (t^i(0) \times t^i(\varepsilon)) = t^i(0) \times t^i(\varepsilon)$



Consideremos la aplicación resultante de concatenar los transportes paralelos, dada por

$$R^{i-1}(\varepsilon) = \left(\tilde{P}^{i-1}(\varepsilon) \right)^T \circ \left(P_i(\varepsilon) \right)^T \circ \tilde{P}^i(\varepsilon) \circ P_i(0)$$

Como el transporte paralelo corresponde a una rotación, se tiene que

$$(P(\varepsilon))^T = (P(\varepsilon))^{-1}$$

Podemos representar esto mediante un diagrama:

$$\begin{array}{ccc}
 \mathbf{t}^{i-1}(\varepsilon) & \longrightarrow & \mathbf{t}^i(\varepsilon) \\
 \uparrow & \psi_{i-1}(\varepsilon) & \uparrow \\
 \mathbf{t}^{i-1}(0) & \longrightarrow & \mathbf{t}^i(0)
 \end{array}$$

Observamos que la aplicación $\mathbf{R}^{i-1}(\varepsilon)$ deja invariante al vector $\mathbf{t}^{i-1}(0)$ y es una composición de rotaciones, con lo que necesariamente se trata de una rotación de ángulo $\psi_{i-1}(\varepsilon)$ alrededor del eje definido por $\mathbf{t}^{i-1}(0)$. Este ángulo $\psi_{i-1}(\varepsilon)$ es lo que llamaremos holonomía de conexión inducida por el transporte paralelo alrededor del giro entre dos puentes consecutivos. Podemos expresar la variación de $\psi_{i-1}(\varepsilon)$ respecto a una variación de la curva $\delta \mathbf{x}$ [de Vries 2005] como

$$\delta \psi_{i-1} = \frac{-2\mathbf{t}^{i-1} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1}\mathbf{t}^i} \left(\frac{1}{2} \frac{\delta \mathbf{x}_i - \delta \mathbf{x}_{i-1}}{\|\mathbf{eR}^{i-1}\|} + \frac{1}{2} \frac{\delta \mathbf{x}_{i+1} - \delta \mathbf{x}_i}{\|\mathbf{eR}^i\|} \right)$$

Recordemos que en el caso continuo, la holonomía de conexión venía dada por

$$\delta \left(\int \psi ds \right) = - \int k \mathbf{b}(\delta \mathbf{x})_s ds$$

Si comparamos ambas expresiones, concluimos que el segundo factor de la igualdad discreta es la aproximación de $(\delta \mathbf{x})_s$, y el primer factor por tanto es el opuesto de la curvatura binormal discreta. El cálculo del gradiente de la holonomía de conexión discreta podemos obtenerlo a partir del segundo factor, ya que éste nos da el incremento según las direcciones, de manera que

$$\begin{aligned}
 \nabla_{i-1} \psi_{i-1} &= \frac{\partial \psi_{i-1}}{\partial \mathbf{x}_{i-1}} = \frac{1}{2} \frac{\frac{2\mathbf{t}^{i-1} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1}\mathbf{t}^i}}{\|\mathbf{eR}^{i-1}\|}, \quad \nabla_{i+1} \psi_{i-1} = \frac{\partial \psi_{i-1}}{\partial \mathbf{x}_{i+1}} = \frac{1}{2} \frac{\frac{-2\mathbf{t}^{i-1} \times \mathbf{t}^i}{1 + \mathbf{t}^{i-1}\mathbf{t}^i}}{\|\mathbf{eR}^i\|}, \\
 \nabla_i \psi_{i-1} &= \frac{\partial \psi_{i-1}}{\partial \mathbf{x}_i} = -(\nabla_{i-1} + \nabla_{i+1}) \psi_{i-1}
 \end{aligned}$$

En el siguiente apartado simplificaremos estos resultados con nuestra expresión del vector de curvatura binormal discreta.

Energía discreta de flexión

En el caso continuo, la energía de flexión de la cuerda venía determinada por la curvatura. Para hallar el análogo a la curvatura en el caso discreto, tendremos que definir un concepto de curvatura discreta sobre nuestra colección de puntos, y nos serviremos del vector de la curvatura binormal ($k\mathbf{b} = \mathbf{t} \times \mathbf{k}$) del caso continuo, cuyo módulo coincidía con la curvatura. Definimos la curvatura binormal discreta en un vértice \mathbf{x}_i como

$$(k\mathbf{b})_{i-1} = \frac{2\mathbf{e}^{i-1} \times \mathbf{e}^i}{\|\mathbf{eR}^{i-1}\| \|\mathbf{eR}^i\| + \mathbf{e}^{i-1} \mathbf{e}^i}$$

Este vector es ortogonal al plano osculador que pasa por dos puentes consecutivos, análogamente al caso continuo, y se observa que nuestra definición del vector $(k\mathbf{b})_{i-1}$ es equivalente a $2\mathbf{t}^{i-1} \times \mathbf{t}^i / (1 + \mathbf{t}^{i-1} \mathbf{t}^i)$, obtenida con la holonomía de conexión. Con esta nueva expresión,

podemos simplificar las expresiones del gradiente de holonomía de conexión, de manera que

$$\nabla_{i-1} \psi_{i-1} = \frac{1}{2} \frac{(k\mathbf{b})_{i-1}}{\|\mathbf{eR}^{i-1}\|}, \nabla_{i+1} \psi_{i-1} = -\frac{1}{2} \frac{(k\mathbf{b})_{i-1}}{\|\mathbf{eR}^i\|}, \nabla_i \psi_{i-1} = -(\nabla_{i-1} + \nabla_{i+1}) \psi_{i-1}$$

Sea ϕ_i el ángulo formado por los dos puentes consecutivos $\mathbf{e}^{i-1}, \mathbf{e}^i$. Si calculamos ahora el módulo de la curvatura binormal, teniendo en cuenta la hipótesis de inextensibilidad, obtenemos que

$$\begin{aligned} \|(k\mathbf{b})_{i-1}\| &= \frac{2\|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| \sin \phi_i}{\|\mathbf{eR}^{i-1}\| \|\mathbf{eR}^i\| + \|\mathbf{e}^{i-1}\| \|\mathbf{e}^i\| \cos \phi_i} = \frac{2 \sin \phi_i}{1 + \cos \phi_i} \\ &= \frac{4 \sin \frac{\phi_i}{2} \cos \frac{\phi_i}{2}}{1 + \cos^2 \frac{\phi_i}{2} - \sin^2 \frac{\phi_i}{2}} = \frac{4 \sin \frac{\phi_i}{2} \cos \frac{\phi_i}{2}}{2 \cos^2 \frac{\phi_i}{2}} = 2 \tan \frac{\phi_i}{2} \end{aligned}$$

Por tanto, el valor de la curvatura discreta para dos puentes consecutivos viene dado por

$$k_{i-1} = 2 \tan \frac{\phi_i}{2}$$

Claramente, como cada uno de los puentes es una recta, el concepto de curvatura discreta está asociado a los vértices de la cuerda y no así a los puentes.

Teniendo ahora tanto el vector curvatura binormal como la curvatura discretizados, podemos dar una expresión discreta de nuestra energía de flexión usando el concepto de cantidad integrada explicado anteriormente.

Recordamos que en el caso continuo para una cuerda recta isotrópica venía dada por

$$E_{\text{flexión}}(\Gamma) = \frac{1}{2} \int \alpha \kappa^2(s) ds$$

Observamos que $k_{i-1}^2 = (k\mathbf{b})_{i-1}^2$, y usando el concepto de cantidad integrada descrita anteriormente podemos escribir

$$E_{\text{flexión}}(\Gamma) = \frac{1}{2} \sum_{i=1}^{n-2} \alpha \left(\frac{(k\mathbf{b})_i}{lR_i/2} \right)^2 \frac{lR_i}{2} = \sum_{i=1}^{n-2} \alpha \frac{(k\mathbf{b})_i^2}{lR_i}$$

Si ahora buscamos la misma expresión para una cuerda no isotrópica, igual que en el caso continuo debemos considerar las curvaturas materiales ω_i^j .

Recordamos que en el caso continuo:

$$\omega = (t'(s)\mathbf{m}_1(s), t'(s)\mathbf{m}_2(s))^T$$

$$E_{\text{flexión}}(\Gamma) = \frac{1}{2} \int (\omega(s) - \overline{\omega(s)})^T \bar{B} (\omega(s) - \overline{\omega(s)}) ds$$

Como ahora no tenemos el vector derivada de la tangente discreta, lo que usaremos será el vector curvatura binormal, y con él podemos expresar la curvatura material discreta como

$$\omega_i^j = ((k\mathbf{b})_i \mathbf{m}_2^j, -(k\mathbf{b})_i \mathbf{m}_1^j) ; j \in \{i-1, i\}$$

La energía de flexión para una cuerda no isotrópica viene dada entonces por

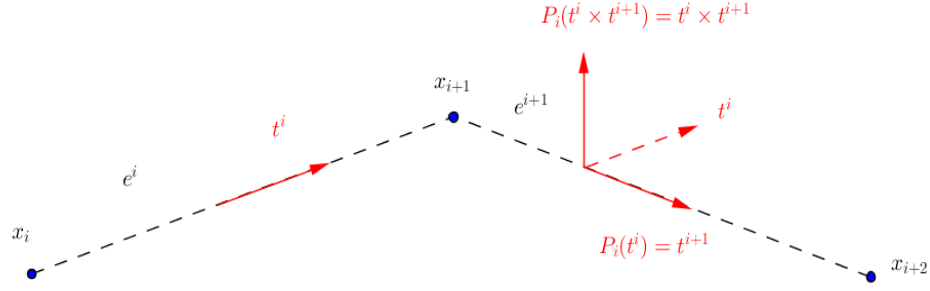
$$E_{\text{flexión}}(\Gamma) = \sum_{i=1}^{n-2} \frac{1}{2lR_i} \sum_{j=i-1}^i (\omega_i^j - \omega \mathbf{R}_i^j)^T \bar{B}^j (\omega_i^j - \omega \mathbf{R}_i^j)$$

Transporte paralelo y sistemas Bishop discretos

De manera análoga al caso continuo, definimos el transporte paralelo discreto como una rotación alrededor del vector curvatura binormal discreto, de manera que

$$P_i(\mathbf{t}^{i-1}) = \mathbf{t}^i, \quad P_i(\mathbf{t}^{i-1} \times \mathbf{t}^i) = \mathbf{t}^{i-1} \times \mathbf{t}^i$$

Observamos que esta rotación no está definida para $\mathbf{t}^{i-1} = -\mathbf{t}^i$, lo que significaría que la cuerda se doblara sobre sí misma.



Para definir los sistemas Bishop discretos, transportamos el vector unitario inicial \mathbf{u}^0 a lo largo de la curva utilizando las rotaciones P_i , de manera que

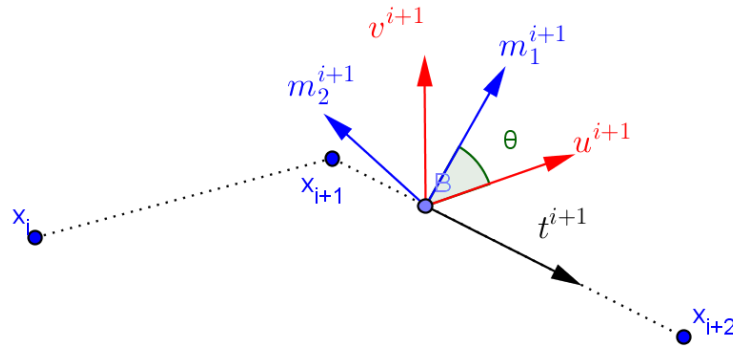
$$\mathbf{u}^i = P_i(\mathbf{u}^{i-1}), \mathbf{v}^i = \mathbf{t}^i \times \mathbf{u}^i$$

Más adelante se demostrará que la elección de este vector inicial unitario \mathbf{u}^0 es arbitraria y no influye en el resultado del cálculo de la energía de torsión siempre y cuando $\mathbf{u}^0 \perp \mathbf{t}^0$

Ahora, podemos representar los sistemas materiales discretos igual que en el caso continuo a partir de

$$\mathbf{m}_1^i = \mathbf{u}^i \cos \theta^i + \mathbf{v}^i \sin \theta^i, \quad \mathbf{m}_2^i = -\mathbf{u}^i \sin \theta^i + \mathbf{v}^i \cos \theta^i$$

Donde θ^i es el ángulo formado por los sistemas Bishop y material i -ésimos.



Energía de torsión

Recordamos que la energía de torsión de la cuerda en el caso continuo venía dada por

$$E_{\text{torsión}}(\Gamma) = \frac{1}{2} \int \beta \, m^2(s) ds = \frac{1}{2} \int \beta \, \theta'^2(s) ds$$

Usando de nuevo la noción de cantidad integrada, podemos escribir

$$\frac{1}{2} \int \beta \, \theta'^2(s) ds \sim \sum_{i=1}^{n-2} \frac{1}{2} \beta \left(\frac{\theta^{i+1} - \theta^i}{lR_i/2} \right)^2 \frac{lR_i}{2} = \sum_{i=1}^{n-2} \beta \frac{(\theta^{i+1} - \theta^i)^2}{lR_i}$$

O equivalentemente:

$$\frac{1}{2} \int \beta \, m^2(s) ds \sim \sum_{i=1}^{n-2} \frac{1}{2} \beta \left(\frac{m_i}{lR_i/2} \right)^2 \frac{lR_i}{2} = \sum_{i=1}^{n-2} \beta \frac{m_i^2}{lR_i}$$

Con lo que escribimos la energía de torsión como

$$E_{\text{torsión}}(\Gamma) = \sum_{i=1}^{n-2} \beta \frac{m_i^2}{lR_i} = \sum_{i=1}^{n-2} \beta \frac{(\theta^{i+1} - \theta^i)^2}{lR_i}$$

Vemos así que la energía de torsión depende del ángulo formado por el sistema material y el sistema Bishop en cada puente. Nuestra elección de nuestro sistema Bishop ha sido arbitraria, así que tendremos que demostrar que realmente no importa la elección inicial del sistema para el cálculo de la energía.

Supongamos que no escogemos correctamente el sistema Bishop inicial (algo bastante plausible teniendo en cuenta que sólo se le ha exigido al vector perpendicularidad a la tangente y ser unitario). Sabemos que existe un sistema Bishop correcto que cumple todas las hipótesis descritas en la teoría continua, y que éste incluye también al vector tangente, con lo que el sistema Bishop correcto es una rotación de nuestro sistema Bishop incorrecto alrededor de la tangente de un ángulo que llamaremos θ^* . Sea θ el ángulo entre el sistema material y el sistema Bishop erróneo.

El ángulo entre el sistema material y el sistema Bishop verdadero vendrá dado por

$$\Theta = \theta \pm \theta^*$$

Debido a que el transporte paralelo es una rotación que tiene por eje la curvatura binormal, este ángulo de error θ^* se mantiene a lo largo de la cuerda, independientemente de la posición en la que nos encontramos.

Sabemos que la energía de torsión viene dada por el ángulo de rotación que transforma el sistema material en el Bishop en cada puente, y más concretamente, por la suma de los cuadrados de sus sucesivas diferencias.

Así, la energía real de torsión dependerá de los sumandos $\Theta^i - \Theta^{i-1}$. Si expandimos este sumando, obtenemos

$$\Theta^i - \Theta^{i-1} = (\theta \pm \theta^*)^i - (\theta \pm \theta^*)^{i-1} = \theta^i \pm \theta^* - \theta^{i-1} \mp \theta^* = \theta^i - \theta^{i-1}$$

Con lo que la energía no depende del sistema Bishop inicial que escojamos.

Condiciones de contorno e hipótesis previas

Para simplificar todo usamos que las ondas de flexión son un término diferencial de segundo orden de nuestro problema. La velocidad de las ondas de torsión aumenta mientras la inercia rotacional de la sección transversal desaparece. Las ondas de flexión resultan ser dispersivas [Sommerfeld 1964], es decir, su velocidad depende de su longitud de onda λ , y viene dada por $v_s h / \lambda$, donde h es el radio de la cuerda y v_s es la velocidad del sonido sobre el material

de la cuerda. Para longitudes de onda tales que $\lambda \gg h$, es decir, mucho mayores que el radio de la cuerda, las ondas de torsión viajan mucho más rápido que las de flexión. En los problemas que vamos a considerar la longitud de onda tomará valores muy altos, con lo que simplificaremos el problema omitiendo la resolución temporal de las ondas de flexión.

Consideramos ahora el límite de la inercia rotacional de una sección transversal que desaparece. Las ondas de torsión se propagan instantáneamente, y en cada instante de tiempo los sistemas materiales son los minimizadores de la energía elástica, sujetos a alguna condición de contorno, con lo que $\partial E(\Gamma) / \partial \theta^j = 0$.

El valor del ángulo θ^j para el eje j -ésimo puede ser dado por condiciones de contorno sobre los sistemas materiales. Consideraremos la cuerda libre de tensiones si no imponemos ninguna condición de contorno, y sujeta si fijamos los sistemas materiales de los puentes inicial y final. Para el resto de ángulos a calcular utilizaremos que $\partial E(\Gamma)/\partial \theta^j = 0$.

Cálculo de ángulos y actualización de la cuerda

Hemos visto que la clave para el cálculo de las energías que actúan en la cuerda es hallar los ángulos θ^j entre los sistemas materiales y Bishop. Distinguiremos dos casos distintos de cuerdas, el primero es en el que la cuerda es isotrópica y recta en el reposo, y el segundo en el que no. Los cálculos del primer caso son mucho más fáciles con lo que conviene hacerlo por separado.

En el caso en el que la cuerda es isotrópica y recta en su estado natural de reposo, la cuerda sufre una torsión uniforme si sus extremos están sujetos. Esto implica que

$$\frac{m_i}{lR_i} = \frac{\theta^{n-1} + \theta^1}{2LR} = cte$$

Con $2LR = \sum_{i=2}^{n-1} lR_i$. Si sustituimos entonces este nuevo valor en la energía de torsión, nos queda

$$E_{\text{torsión}}(\Gamma) = \sum_{i=1}^{n-2} \beta \frac{m_i^2}{lR_i} = \sum_{i=1}^{n-2} \beta \left(\frac{m_i}{lR_i} \right)^2 lR_i = \beta \left(\frac{\theta^{n-1} + \theta^1}{2LR} \right)^2 \sum_{i=1}^{n-2} lR_i = \beta \frac{(\theta^{n-1} + \theta^1)^2}{2LR}$$

Con lo que la energía total será

$$E(\Gamma) = \sum_{i=1}^{n-2} \alpha \frac{(kb)_i^2}{lR_i} + \beta \frac{(\theta^{n-1} + \theta^1)^2}{2LR}$$

Si consideráramos condiciones de contorno en las que la cuerda está libre de tensiones, no habría energía de torsión.

En el caso general de una cuerda no isotrópica usaremos el método de newton para minimizar la energía elástica con respecto a los sistemas materiales, que requiere tanto el gradiente como la Hessiana de la energía respecto a la variable θ^j . El gradiente viene dado por

$$\frac{\partial E(\Gamma)}{\partial \theta^j} = \frac{\partial}{\partial \theta^j} (W_j + W_{j+1}) + 2\beta \left(\frac{m_j}{lR_j} - \frac{m_{j+1}}{lR_{j+1}} \right)$$

Donde

$$\frac{\partial}{\partial \theta^j} W_i = \frac{1}{lR_i} (\omega_i^j)^T J \bar{B}^j (\omega_i^j - \omega R_i^j)$$

Con J una rotación de ángulo $\frac{\pi}{2}$ en contra del sentido de las agujas del reloj que actúa sobre vectores de 2D. Nótese que en este caso la energía de flexión también depende indirectamente del ángulo θ^j .

La Hessiana H se obtiene derivando $\frac{\partial E(\Gamma)}{\partial \theta^j}$ en función de $\theta^{j-1}, \theta^j, \theta^{j+1}$, y se obtiene

$$H_{j,j-1} = -\frac{2\beta}{lR_j}, H_{j,j+1} = -\frac{2\beta}{lR_{j+1}}$$

$$H_{j,j} = \frac{2\beta}{lR_j} + \frac{2\beta}{lR_{j+1}} + \frac{\partial^2}{(\partial \theta^j)^2} (W_j + W_{j+1})$$

Donde

$$\frac{\partial^2}{(\partial \theta^j)^2} W_i = \frac{1}{lR_i} (\omega_i^j)^T J^T \bar{B}^j J (\omega_i^j) - \frac{1}{lR_i} (\omega_i^j)^T \bar{B}^j (\omega_i^j - \omega R_i^j)$$

Ecuaciones del movimiento

Nuestro objetivo final es encontrar las ecuaciones de movimiento de la cuerda, que en analogía al caso continuo vienen dadas por

$$M \ddot{\mathbf{x}} = -\frac{dE(\Gamma)}{d\mathbf{x}}$$

Donde M es una matriz diagonal de tamaño $3n \times 3n$ que asocia una masa a cada uno de los vértices de nuestra cuerda discreta. Desarrollemos el segundo término vértice a vértice.

$$-\frac{dE(\Gamma)}{d\mathbf{x}_i} = -\frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} - \sum_{j=1}^{n-1} \frac{\partial E(\Gamma)}{\partial \theta^j} \frac{\partial \theta^j}{\partial \mathbf{x}_i}$$

Se observa que esta ecuación varía enormemente en función de las condiciones de contorno. Si la cuerda es libre de tensiones, se tiene que todo ángulo cumple $\partial E(\Gamma)/\partial \theta^j = 0$,

con lo que el sumatorio sería 0, y tendríamos

$$-\frac{dE(\Gamma)}{d\mathbf{x}_i} = -\frac{\partial E(\Gamma)}{\partial \mathbf{x}_i}$$

En caso de una cuerda sujeta, el único sumando no nulo del sumatorio es el relativo al puente final, con lo que la igualdad la reexpresamos como

$$-\frac{dE(\Gamma)}{dx_i} = -\frac{\partial E(\Gamma)}{\partial x_i} + \frac{\partial E(\Gamma)}{\partial \theta^{n-1}} \frac{\partial \theta^{n-1}}{\partial x_i}$$

Para calcular ahora el término $\frac{\partial \theta^{n-1}}{\partial x_i}$, necesitaremos introducir una nueva herramienta, a la que llamaremos variación del transporte paralelo.

Usando el concepto de holonomía discreta podemos obtener el cambio en el sistema Bishop cuando variamos la curva que representa el centro de la cuerda. Nos interesa el ángulo que rota el sistema Bishop alrededor de la tangente. Podemos dibujar un diagrama de la situación igual que hicimos en el caso de la holonomía:

$$\begin{array}{ccccccc} \mathbf{F}^0(\varepsilon) & \xrightarrow{0} & \mathbf{F}^1(\varepsilon) & \xrightarrow{0} & \dots & \mathbf{F}^{j-1}(\varepsilon) & \xrightarrow{0} & \mathbf{F}^j(\varepsilon) \\ 0 \uparrow & \psi_1(\varepsilon) & \uparrow & \psi_2(\varepsilon) & & \uparrow & \psi_j(\varepsilon) & \uparrow \Psi^j \\ \mathbf{F}^0(0) & \xrightarrow{0} & \mathbf{F}^1(0) & \xrightarrow{0} & \dots & \mathbf{F}^{j-1}(0) & \xrightarrow{0} & \mathbf{F}^j(0) \end{array}$$

Aquí, escribimos los sistemas Bishop \mathbf{F}^i en vez de las tangentes para mostrar que se les está aplicando un transporte paralelo. Cada flecha representa el ángulo $\psi_i(\varepsilon)$ necesario para alinear un sistema Bishop con el posterior. Las flechas remarcadas con un 0 indican que no se requiere torsión para alinear \mathbf{F}^{i-1} con \mathbf{F}^i . Nos interesa el ángulo final de rotación Ψ^j , requerido para alinear $\tilde{\mathbf{P}}^j(\varepsilon) \left(\mathbf{F}^j(0) \right)$ con $\mathbf{F}^j(\varepsilon)$.

Como el transporte paralelo conmuta con respecto a la torsión y la holonomía es aditiva bajo concatenación de giros, podemos expresar el gradiente del ángulo buscado como

$$\nabla_i \Psi^j = \sum_{k=1}^j \nabla_i \psi_k$$

Entonces, perturbar la curva rota el sistema Bishop en el puente \mathbf{e}^n un ángulo Ψ^n alrededor de la tangente \mathbf{t}^n . Así, para hacer que el sistema material del último puente cumpla las condiciones de contorno debemos sustraer este ángulo del ángulo final θ^n , obteniendo

$$-\frac{dE(\Gamma)}{dx_i} = -\frac{\partial E(\Gamma)}{\partial x_i} + \frac{\partial E(\Gamma)}{\partial \theta^{n-1}} \sum_{j=1}^{n-1} \frac{\partial \psi_j}{\partial x_i}$$

Expresamos este resultado ahora tanto para el caso de una cuerda isotrópica recta como el general.

En el caso de la cuerda isotrópica recta, las fuerzas sobre el vértice \mathbf{x}_i vienen dadas por tres contribuciones:

$$-\frac{2\alpha}{lR_j}(\nabla_i(\mathbf{k}\mathbf{b})_j)^T(\mathbf{k}\mathbf{b})_j + \frac{\beta(\theta^{n-1} - \theta^1)}{LR}\nabla_i\psi_j, i-1 \leq j \leq i+1$$

Con

$$\begin{aligned}\nabla_{i-1}(\mathbf{k}\mathbf{b})_{i-1} &= \frac{2[\mathbf{e}^i] + (\mathbf{k}\mathbf{b})_{i-1}(\mathbf{e}^i)^T}{\|\mathbf{e}^{\mathbf{R}^{i-1}}\| \|\mathbf{e}^{\mathbf{R}^i}\| + \mathbf{e}^{i-1}\mathbf{e}^i} \\ \nabla_{i+1}(\mathbf{k}\mathbf{b})_{i-1} &= \frac{2[\mathbf{e}^{i-1}] + (\mathbf{k}\mathbf{b})_{i-1}(\mathbf{e}^{i-1})^T}{\|\mathbf{e}^{\mathbf{R}^{i-1}}\| \|\mathbf{e}^{\mathbf{R}^i}\| + \mathbf{e}^{i-1}\mathbf{e}^i} \\ \nabla_i(\mathbf{k}\mathbf{b})_{i-1} &= -(\nabla_{i-1} + \nabla_{i+1})(\mathbf{k}\mathbf{b})_{i-1}\end{aligned}$$

Donde $[\mathbf{e}^i]$ es una matriz que actúa sobre los vectores 3D que cumple que $[\mathbf{e}^i]\mathbf{x} = \mathbf{e}^i \times \mathbf{x}$.

El caso general resulta más complicado. Las expresiones de las fuerzas en la cuerda vienen dadas por

$$\begin{aligned}\frac{\partial E(\Gamma)}{\partial \theta^{n-1}} &= \frac{1}{lR_{n-2}}(\boldsymbol{\omega}_{n-2}^{n-1})^T J \bar{B}^n(\boldsymbol{\omega}_{n-2}^{n-1} - \boldsymbol{\omega}^{\mathbf{R}_{n-2}^{n-1}}) + 2\beta \frac{m_{n-2}}{lR_{n-2}} \\ \frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} &= \sum_{k=1}^{n-2} \frac{1}{lR_k} \sum_{j=k}^{k+1} (\nabla_i \boldsymbol{\omega}_k^j)^T \bar{B}^j(\boldsymbol{\omega}_k^j - \boldsymbol{\omega}^{\mathbf{R}_k^j}) \\ \sum_{j=1}^{n-1} \frac{\partial \psi_j}{\partial \mathbf{x}_i} &= \sum_{j=i-2}^i \frac{\partial \psi_j}{\partial \mathbf{x}_i} = \sum_{j=i-2}^i \nabla_i \psi_j\end{aligned}$$

Donde el gradiente de la curvatura del sistema material $\boldsymbol{\omega}_k^j$ viene dado por

$$\nabla_i \boldsymbol{\omega}_k^j = \begin{pmatrix} (\mathbf{m}_2^j)^T \\ -(\mathbf{m}_1^j)^T \end{pmatrix} \nabla_i(\mathbf{k}\mathbf{b})_k - J \boldsymbol{\omega}_k^j (\nabla_i \Psi^j)^T$$

Se observa que para el primer y último vértice la suma $\sum_{j=i-2}^i \nabla_i \psi_j$ tiene un solo sumando, $\nabla_1 \psi_1$ y $\nabla_n \psi_{n-2}$ respectivamente, y dos para el segundo y el penúltimo vértice, que son $\nabla_2 \psi_1$ $\nabla_2 \psi_2$ y $\nabla_{n-1} \psi_{n-3}$ $\nabla_1 \psi_{n-2}$.

Ya tenemos todo el desarrollo teórico necesario para resolver las ecuaciones del movimiento que nos darán la solución del problema. Usaremos el método de proyección múltiple VERLET para forzar la inextensibilidad de la cuerda.

Capítulo 2 - Simulación de la cuerda elástica discreta

Nuestro objetivo es ver cómo evoluciona una cuerda elástica sometida a distintas fuerzas externas e internas dadas unas condiciones iniciales. El estudio y las demostraciones teóricas se han incluido anteriormente, y este capítulo se centrará en la implementación del método, independientemente de que se entienda en qué subyace.

Nuestra cuerda en la simulación vendrá dada por un conjunto de puntos tridimensionales, y por los puentes o aristas que los unen. Supondremos que es inextensible para simplificar el problema. Esto se forzará mediante restricciones en vez de aplicarlo directamente a la teoría para no complicar el problema.

La ecuación diferencial discretizada que tendremos que resolver será la siguiente:

$$M\ddot{\mathbf{x}} = -\frac{dE(\Gamma)}{d\mathbf{x}}$$

Donde M es la matriz de masas de la cuerda, \mathbf{x} la posición de los vértices de la cuerda discretizada y $E(\Gamma)$ la energía de la misma. Éste último término es el más complicado de todos, ya que dependerá de numerosos parámetros que bien tendremos que introducir o bien tendremos que calcular a partir de la cuerda dada. La fórmula de la derivada de la energía en uno de los vértices viene dada, bajo ciertas condiciones de contorno (ver capítulo 1), por

$$-\frac{dE(\Gamma)}{d\mathbf{x}_i} = -\frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} + \frac{\partial E(\Gamma)}{\partial \theta^{n-1}} \frac{\partial \theta^{n-1}}{\partial \mathbf{x}_i}$$

Y a su vez, para una cuerda con extremos sujetos:

$$\begin{aligned} \frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} &= \sum_{k=1}^{n-2} \frac{1}{l_{R_k}} \sum_{j=k}^{k+1} (\nabla_i \omega_k^j)^T \bar{B}^j (\omega_k^j - \omega_{R_k}^j) \\ \frac{\partial E(\Gamma)}{\partial \theta^{n-1}} &= \frac{1}{l_{n-1}} (\omega_{n-2}^{n-2})^T J \bar{B}^n (\omega_{n-2}^{n-2} - \omega_{R_{n-2}}^{n-2}) + 2\beta \frac{\theta^{n-1} - \theta^1}{l_{n-1}} \end{aligned}$$

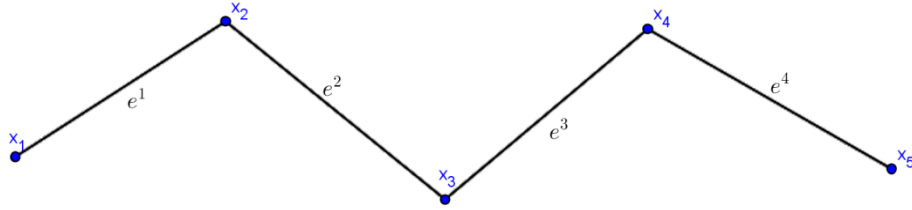
El significado de todas estas variables puede verse en el capítulo 1, pero no hay que perder de vista que nuestros dos objetivos son primero hallar esos valores y segundo resolver el sistema de ecuaciones del movimiento por un método numérico. Nosotros usaremos el método Verlet.

Introducción de los datos iniciales y en el equilibrio

Por comodidad para la implementación del algoritmo, conviene introducir al principio del código ciertos parámetros variables como el tamaño del paso temporal o los ejes donde se quiera representar la cuerda.

Consideraremos una cuerda discretizada representada por un conjunto de puntos o vértices $x_i, i \in \{1, \dots, n\}$ y un conjunto de puentes entre los mismos

$e^i, i \in \{1, \dots, n-1\} / e^i = x_{i+1} - x_i$, tal como muestra la siguiente figura:



Lo primero que debemos hacer es introducir el número de vértices n de los que se compondrá la cuerda. A un mayor número de estos, la simulación será más precisa, ya que la cuerda discreta resultante será menos rígida, pero también aumentará el coste computacional, debiendo bajar también el tamaño del paso temporal para obtener resultados fiables.

Para realizar la simulación también será necesaria la posición de nuestra cuerda en el equilibrio, es decir, a qué configuración es a la que va a tender la cuerda. Debemos introducir la posición de los n vértices en el equilibrio, en un vector de tamaño $3n$ al que llamaremos XR .

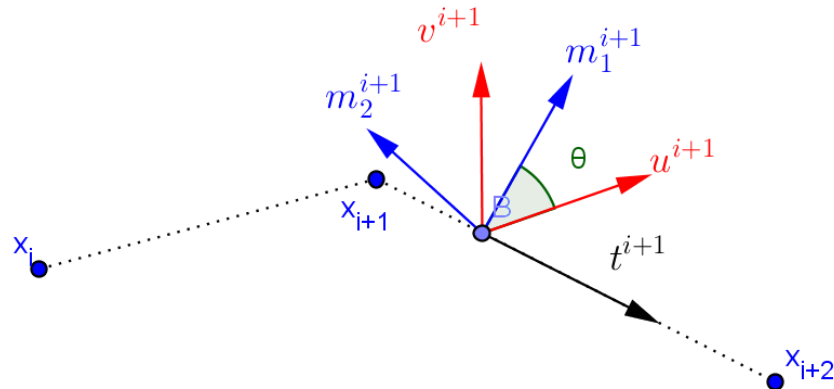
También debemos introducir la posición de los vértices iniciales. Ésta debe de estar en un vector columna de dimensión $3n$, de manera que los valores $[3k+1, 3k+2, 3(k+1)]$ correspondan a la posición tridimensional $[x, y, z]$ de nuestro punto $(k+1)$ -ésimo. Tenemos así guardadas las posiciones iniciales de la cuerda en un vector X . Conviene que estas posiciones iniciales no estén demasiado perturbadas para que se cumpla la hipótesis de inextensibilidad de la cuerda. De igual

modo, introducimos las velocidades iniciales de la cuerda, que también deben de estar en un único vector columna de dimensión $3n$ al que llamaremos V . También podemos añadir una fuerza externa F_{ext} que actúe sobre cada vértice de la cuerda.

Lo siguiente que debemos hacer es introducir los parámetros físicos de la cuerda: el de flexión, el de torsión y la matriz de masas asociada. Para simplificar cálculos se considerará que la densidad de masa de la cuerda es uniforme, con lo que la matriz será diagonal con el elemento en la diagonal constante. Al parámetro de flexión lo llamaremos α , al de torsión β y a la matriz de masas M .

Dada ya la posición de los vértices iniciales y la de los vértices en reposo, hay que calcular los $n-1$ puentes entre los mismos. A la matriz de tamaño $n-1 \times 3$ con los puentes que unen los vértices en el reposo la llamaremos eR , mientras que a la matriz de tamaño $n-1 \times 3$ con los puentes que unen los vértices iniciales la llamaremos e . El cálculo de los puentes se realiza sin más que restarle a cada vértice su anterior. También calculamos la medida de cada par de puentes consecutivos tanto en reposo como en el inicio, en dos vectores de tamaño $n-2$ que llamaremos lR y l respectivamente. La explicación del porqué de cada par de puentes consecutivos en vez de los mismos puentes se encuentra en el primer capítulo (página 12, cantidad integrada).

Las características físicas de la cuerda vendrán descritas por unos sistemas ortonormales definidos en cada uno de los puentes, que medirán las energías de torsión y de flexión. En cada uno de los puentes se definirán tanto un sistema material, $\{t^i, m_1^i, m_2^i\}$, como un sistema Bishop, $\{t^i, u^i, v^i\}$, que como se explica en los apéndices es el que tiene “torsión nula”. El cálculo de las energías de la cuerda vendrá dado en función del ángulo θ formado entre ambos.



Introducimos ahora el sistema Bishop del primer puente, al que llamaremos $b0$. Los sistemas Bishop son sistemas ortonormales que se asocian a cada puente de la cuerda, que incluyen al vector unitario t^i con dirección y sentido el del puente sobre el que está definido. La elección de este sistema es arbitraria, pero es imprescindible que sean ortonormales.

Por último, queda incluir las condiciones de contorno de la cuerda. En nuestro caso, supondremos que está sujeta, es decir, debemos dar un par de sistemas ortonormales, a los que llamaremos sistemas materiales, tanto en el primer puente como en el último en el que se incluyan los vectores unitarios con dirección y sentido del puente en el que se definan. De este modo, al sistema material del primer puente lo llamaremos mf_i , y al del segundo mf_f . De nuevo, es imprescindible se sean ortonormales.

Una manera alternativa más útil de hacer esto último podría ser dar las condiciones de contorno en función del ángulo θ existente entre los sistemas Bishop y material del primer y último puente. También se pueden añadir parámetros para que la variación de las condiciones de contorno en el tiempo.

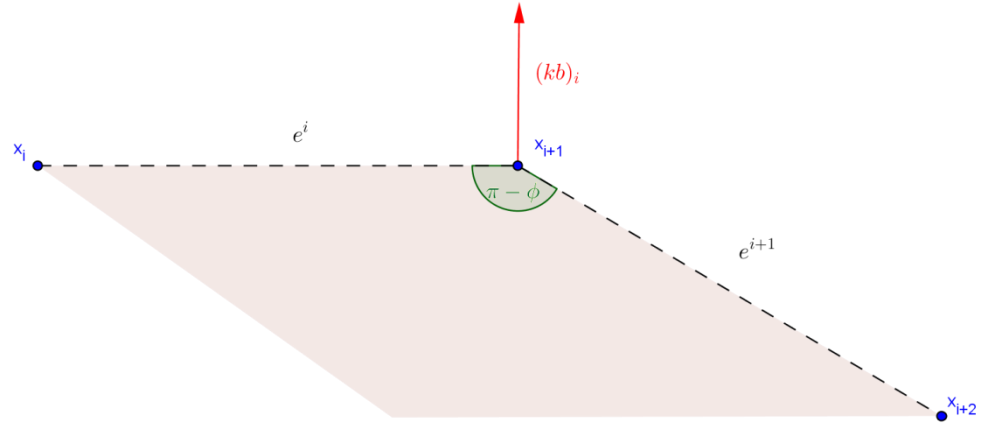
Construcción de los datos en el equilibrio

A partir de los datos introducidos toca ahora calcular ciertos valores que se derivan de ellos y que servirán para la resolución del problema. Todos ellos miden propiedades físicas de la cuerda y servirán para resolver sus ecuaciones discretas del movimiento más adelante. En este apartado nos limitaremos a calcular los valores relacionados con la posición en el equilibrio. Toda explicación teórica de los mismos se explica en el capítulo 1.

Comenzamos calculando los vectores curvatura binormal en el equilibrio, a los que llamaremos kbR . Estos vectores son ortogonales al plano formado por dos puentes consecutivos, y servirán para calcular los sistemas Bishop en reposo de cada uno de los puentes. La curvatura binormal en reposo viene dada por

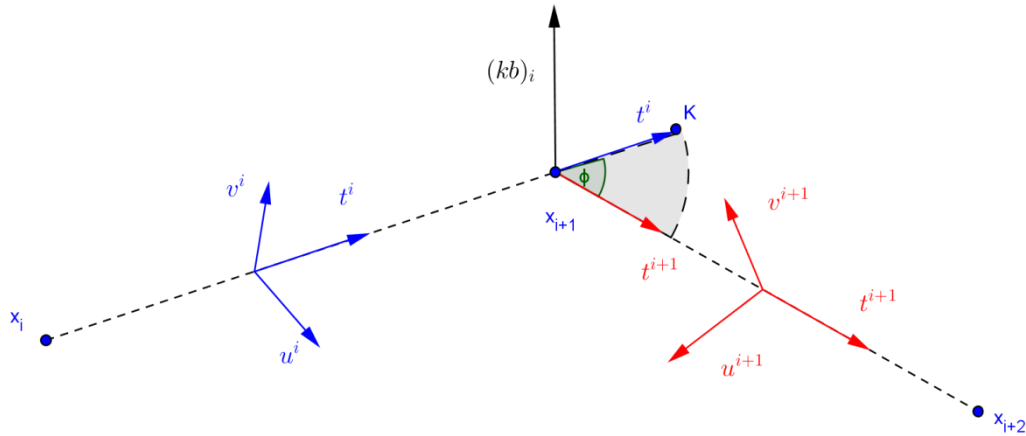
$$(kbR)_i = \frac{2eR^i \times eR^{i+1}}{\|eR^{i+1}\| \|eR^i\| + eR^{i+1} eR^i}, i = 1 \dots n - 2$$

Estos vectores se definen en cada v rtice y son ortogonales al plano formado por los dos puentes confluyentes en el v rtice en el que se define, tal como se indica en la figura:



Se utilizar n para aplicar el concepto de transporte paralelo y para calcular las energ as de la cuerda.

Ahora podemos calcular la posici n de los sistemas Bishop en reposo, que servir n para calcular los sistemas materiales en reposo que a su vez nos dar n par metros necesarios para el c lculo de energ as. Para ello haremos uso de un concepto te rico llamado transporte paralelo, que b sicamente se trata de una rotaci n del sistema Bishop inicial a lo largo de la curva. Rotamos el sistema de un puente a otro alrededor del vector curvatura binormal que forman ambos. El  ngulo de rotaci n viene dado por el  ngulo formado entre los vectores que forman ambos puentes, ϕ_i . Para hallarlo no hay m s que calcular el producto escalar de ambos vectores y despejar el  ngulo mediante el arcocoseno. Hay que fijarse que  ste es el menor  ngulo formado entre ambos puentes, y puede ser que no corresponda al giro que solapa eR^i sobre eR^{i+1} , con lo que simplemente hay que comprobar despu s que esto es as  rot ndolo. En caso de que no sea, se tomar  $-\phi_i$



Para calcular la rotación que supone el transporte paralelo se pueden usar cuaterniones. Calculamos entonces una matriz de dimensión $3 \times 3 \times n-1$ a la que llamaremos *bishopR* en la que se guardarán las $n-1$ matrices que contienen los sistemas Bishop en el equilibrio de cada uno de los $n-1$ puentes de la cuerda.

Lo siguiente a calcular son los ángulos de giro entre los sistemas materiales y los sistemas Bishop en el equilibrio. A esta variable la llamaremos *thetaR*, y será un vector que contendrá los $n-1$ ángulos de giro. Aquí, dependiendo del comportamiento físico de la cuerda, tenemos dos posibilidades: la primera que la cuerda sea isotrópica y recta en el equilibrio y la segunda que no lo sea. Isotrópica significa que no favorece ninguna dirección de giro por encima de las demás. En el primer caso, la fórmula que determina los ángulos viene dada por

$$\frac{\theta R^{i+1} - \theta R^i}{lR_i} = \frac{\theta R^{n-1} - \theta R^1}{2LR} = cte$$

Donde $2LR$ es la longitud total de la cuerda y θR^i es el ángulo de giro asociado al giro entre el sistema bishop y el material del puente i -ésimo. Como nuestras condiciones de contorno son para una cuerda sujeta, podemos calcular el ángulo necesario para convertir los sistemas Bishop del primer y último puente en los materiales, y así obtener el valor constante que toma el cociente más arriba descrito. Después, simplemente despejando en la ecuación obtenemos los ángulos restantes.

En el caso en el que la cuerda no sea isotrópica y de posición recta en el equilibrio el cálculo de los ángulos se complica.

Los ángulos en el reposo vendrían dados por

$$0 = \frac{\theta_{R^{j+1}} - \theta_{R^j}}{l_{R_j}} - \frac{\theta_{R^{j+2}} - \theta_{R^{j+1}}}{l_{R_{j+1}}}, j = 1 \dots n-3$$

De igual modo que en el caso isotrópico, conocemos los ángulos inicial y final, con lo que obtenemos un sistema de $n-2$ ecuaciones con $n-2$ incógnitas con el que podemos hallar todos los valores de nuestro vector θ_{R^j} . En cualquiera de los casos, tenemos que dar unas condiciones de contorno en el equilibrio para los valores de los ángulos θ_{R^j} en los puentes extremos-

Una vez calculamos los ángulos de giro entre sistemas Bishop y material en cada uno de los puentes, hallamos el sistema material de los puentes en el reposo, sin más que rotar alrededor de la tangente el ángulo obtenido en el apartado anterior. Obtenemos así una matriz de tamaño $3 \times 3 \times n-1$ a la que llamaremos $mframeR$, que de igual modo que la matriz $bishopR$ guardará los sistemas materiales de los puentes.

Ahora calculamos la curvatura material discreta de la cuerda en el equilibrio. Ésta viene dada por

$$\omega R_i^j = ((kbR)_i mR_2^j, -(kbR)_i mR_1^j) ; j \in \{i+1, i\}$$

Donde mR_1^j, mR_2^j son los dos vectores restantes del sistema material del puente j -ésimo. Así, debemos una matriz de tamaño $n-2 \times 2 \times 2$ a la que llamaremos wR , y en la que guardaremos los sucesivos vectores de curvatura material discreta, de manera que $wR(k, i, :)$ corresponde a $\omega R_k^i = ((kbR)_k mR_2^i, -(kbR)_k mR_1^i)$. Se puede observar que este valor no es más que la proyección del vector $(kbR)_i$ sobre los sistemas materiales de cada uno de los dos puentes consecutivos.

Por último, calculamos las medidas de los puentes en el reposo para más adelante usarlas para imponer nuestra restricción de inextensibilidad. A este vector con las $n-2$ medidas de los puentes lo llamaremos *medidas*.

Construcción de los datos iniciales

Del mismo modo que construimos los datos en el equilibrio, tenemos ahora que construir los datos iniciales del problema. Empezamos con los vectores iniciales de la curvatura binormal.

De manera análoga al caso del equilibrio, viene dada por

$$(kb)_i = \frac{2\mathbf{e}^i \times \mathbf{e}^{i+1}}{\|\mathbf{e}R^{i+1}\| \|\mathbf{e}R^i\| + \mathbf{e}^{i+1}\mathbf{e}^i}, i = 1 \dots n-2$$

Introducimos estos vectores en una matriz de tamaño $n-2 \times 3$ a la que llamaremos kb .

Igual que en el caso del equilibrio, ahora podemos calcular los sistemas Bishop iniciales a lo largo de la cuerda mediante el transporte paralelo. Guardamos estos vectores en una matriz de tamaño $3 \times 3 \times n-1$ a la que llamaremos $bishop$.

Después de esto calculamos los ángulos de giro entre el sistema material y el Bishop iniciales, de igual modo en que lo hicimos en el caso del equilibrio. Los guardamos en un vector de dimensión $n-1$ al que llamaremos $theta$. Usamos la misma fórmula si se trata de una cuerda isotrópica de configuración recta en el equilibrio:

$$\frac{\theta^{i+1} - \theta^i}{lR_i} = \frac{\theta^{n-1} - \theta^1}{2LR} = cte$$

Si se trata del caso general, tendremos que usar el método de Newton para hallar los θ^j (ver Teoría discreta, cálculo de ángulos y actualización de la cuerda).

Calculamos ahora los sistemas materiales iniciales $\mathbf{m}_1^i, \mathbf{m}_2^i$ y los guardamos en una matriz $3 \times 3 \times n-1$ sin más que rotar los sistemas Bishop el ángulo obtenido anteriormente alrededor de la tangente.

Calculamos ahora la curvatura material discreta inicial ω_k^j , del mismo modo que lo hicimos en el caso del equilibrio, obteniendo una matriz de tamaño $n-2 \times 2 \times 2$.

Ahora tenemos que calcular nuevos valores que en el caso del equilibrio no calculamos. Servirán para hallar el valor de las ecuaciones de movimiento que describen el comportamiento de la cuerda. El primero de ellos será el gradiente del vector curvatura binormal discreta, que se usará para hallar el valor del gradiente de la curvatura material discreta $\nabla_i \omega_k^j$. Éste será una matriz de tamaño $3 \times 3 \times n \times n-2$ al que llamaremos $grad_kb$, de manera que $grad_kb(:, :, i, j)$ será una matriz de tamaño 3×3 que guardará el valor del gradiente sub- i de la curvatura binormal j -ésima. Las fórmulas que dan el valor de los gradientes vienen dadas por

$$\nabla_{i-1}(kb)_i = \frac{2[\mathbf{e}^i] + (kb)_i(\mathbf{e}^i)^T}{\|\mathbf{e}R^{i-1}\| \|\mathbf{e}R^i\| + \mathbf{e}^{i-1}\mathbf{e}^i}$$

$$\nabla_{i+1}(k\mathbf{b})_i = \frac{2[\mathbf{e}^{i-1}] + (k\mathbf{b})_i(\mathbf{e}^{i-1})^T}{\|\mathbf{e}\mathbf{R}^{i-1}\| \|\mathbf{e}\mathbf{R}^i\| + \mathbf{e}^{i-1}\mathbf{e}^i}$$

$$\nabla_i(k\mathbf{b})_i = -(\nabla_{i-1} + \nabla_{i+1})(k\mathbf{b})_i$$

Donde $[\mathbf{e}^i]$ es una matriz que actúa sobre vectores 3D de manera que $[\mathbf{e}^i]\mathbf{x} = \mathbf{e}^i \times \mathbf{x}$.

Otro valor que tenemos que hallar para dar el valor del gradiente de la curvatura material discreta $\nabla_i \omega_k^j$ es el gradiente de la holonomía de conexión, que será una matriz de tamaño $3 \times n \times n - 2$ a la que llamaremos *grad_phi* dada por

$$\nabla_i \psi_i = \frac{1}{2} \frac{(k\mathbf{b})_i}{\|\mathbf{e}\mathbf{R}^i\|}, \nabla_{i+2} \psi_i = -\frac{1}{2} \frac{(k\mathbf{b})_i}{\|\mathbf{e}\mathbf{R}^{i+1}\|}, \nabla_{i+1} \psi_i = -(\nabla_i + \nabla_{i+2})\psi_i, i = 1, \dots, n-2$$

Donde $\text{grad_phi}(:, j, i) = \nabla_j \psi_i$. Los valores de la matriz no definidos arriba son nulos.

A partir de este valor podemos ahora calcular el gradiente del ángulo resultante de giro (ver teoría discreta, ecuaciones del movimiento) que influye directamente en el valor del gradiente de la curvatura material discreta. Se trata de una matriz de tamaño $3 \times n \times n - 1$ a la que llamaremos *grad_angrot* y que viene dada por

$$\nabla_i \Psi^j = \sum_{k=1}^j \nabla_i \psi_k$$

con $\text{grad_angrot}(:, i, j) = \nabla_i \Psi^j$.

Una vez dados estos valores, ya podemos hallar el valor del gradiente de la curvatura material discreta, que viene dado por

$$\nabla_i \omega_k^j = \begin{pmatrix} (\mathbf{m}_2^j)^T \\ -(\mathbf{m}_1^j)^T \end{pmatrix} \nabla_i (k\mathbf{b})_k - J \omega_k^j (\nabla_i \Psi^j)^T$$

Con J una rotación de ángulo $\frac{\pi}{2}$ en contra del sentido de las agujas del reloj que actúa sobre vectores de 2D. Podemos ver que para i, j, k fijados, $\nabla_i \omega_k^j$ es una matriz de tamaño 2×3 , con lo que guardaremos los valores en una matriz de tamaño $2 \times 3 \times n \times n - 2 \times 2$ a la que llamaremos *grad_w*.
 $\text{grad_w}(:, :, i, j, k) = \nabla_i \omega_k^j$

Cálculo de las energías iniciales

Ya tenemos el componente más complicado de la expresión

$$\frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} = \sum_{k=1}^{n-2} \frac{1}{lR_k} \sum_{j=k}^{k+1} (\nabla_i \boldsymbol{\omega}_k^j)^T \bar{B}^j (\boldsymbol{\omega}_k^j - \boldsymbol{\omega} \mathbf{R}_k^j)$$

Con lo que únicamente queda hallar el valor de la derivada de la energía con respecto a cada vértice de la cuerda. \bar{B}^j es la matriz de una forma cuadrática asociada a las características de flexión de la cuerda (ver capítulo 1), y medirá la facilidad que tiene ésta para doblarse en determinadas direcciones. En nuestro caso de cuerda isotrópica, la matriz será $\bar{B}^j = \alpha I_{2 \times 2}$. Guardaremos esta nueva variable como B . Para mayor comodidad esta variable puede también definirse en los parámetros del problema al principio del código.

Calculamos ahora el valor de cada uno de los $\partial E(\Gamma)/\partial \mathbf{x}_i$, y lo guardamos en un vector de tamaño $3n$ al que llamaremos dE_x , de manera que $\partial E(\Gamma)/\partial \mathbf{x}_i = dE_x(3i - 2:3i)$

Ya tenemos el primer valor de los que definen las ecuaciones del movimiento. Recordemos que el segundo viene dado por

$$\frac{\partial E(\Gamma)}{\partial \theta^{n-1}} = \frac{1}{lR_{n-1}} (\boldsymbol{\omega}_{n-2}^{n-2})^T J \bar{B}^n (\boldsymbol{\omega}_{n-2}^{n-2} - \boldsymbol{\omega} \mathbf{R}_{n-2}^{n-2}) + 2\beta \frac{\theta^{n-1} - \theta^1}{lR_{n-1}}$$

Donde J era una rotación de ángulo $\pi/2$ en contra de las agujas del reloj. A este nuevo valor lo llamaremos dE_theta .

Sólo queda hallar el valor que toma $\partial \theta^{n-1}/\partial \mathbf{x}_i$. A partir de la teoría (ver capítulo 1, ecuaciones del movimiento) podemos ver que

$$-\frac{\partial \theta^{n-1}}{\partial \mathbf{x}_i} = \sum_{j=1}^{n-2} \frac{\partial \psi_j}{\partial \mathbf{x}_i} = \sum_{j=i-2}^i \frac{\partial \psi_j}{\partial \mathbf{x}_i} = \sum_{j=i-2}^i \nabla_i \psi_j$$

Al vector de tamaño $3n$ resultado de concatenar cada uno de los valores anteriores para cada vértice lo llamaremos $dphi$. Los sumandos en los que los índices sobrepasan el rango de ψ_j son nulos.

Hecho esto último, podemos hallar el vector $-dE(\Gamma)/d\mathbf{x}$ que es el responsable de las ecuaciones del movimiento. Recordamos que

$$-\frac{dE(\Gamma)}{d\mathbf{x}_i} = -\frac{\partial E(\Gamma)}{\partial \mathbf{x}_i} - \frac{\partial E(\Gamma)}{\partial \theta^{n-1}} \frac{\partial \theta^{n-1}}{\partial \mathbf{x}_i}$$

Con lo que el vector $-dE(\Gamma)/d\mathbf{x}$ es un vector de dimensión $3n$ dado por la concatenación de todas las derivadas de la energía con respecto a los distintos vértices. A este vector lo llamaremos dE .

Proyección múltiple

Nuestro objetivo es resolver las ecuaciones del movimiento dadas por $M\ddot{\mathbf{x}} = -dE(\Gamma)/d\mathbf{x}$ pero a su vez restringiéndolas a la hipótesis de inextensibilidad de la cuerda. Para ello, primero resolveremos el sistema diferencial sin considerar ninguna restricción y luego aplicaremos el Fast projection method para añadir las restricciones.

Lo programaremos en una función externa a la que llamaremos cada vez que tengamos que aplicarlo, es decir, para cada uno de los instantes de tiempo de la simulación.

Consideramos nuestra ecuación diferencial

$$\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

La discretizamos dividiéndola en un sistema de dos ecuaciones de la siguiente forma:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}) \end{cases} \Rightarrow \begin{cases} \mathbf{v}^{n+1} = \mathbf{v}^n + h \mathbf{f}(\mathbf{x}^n) \\ \mathbf{x}^{n+1} = \mathbf{x}^n + h \mathbf{v}^{n+1} \end{cases}$$

De este modo, dando un tamaño de paso h y la función inicial de la cuerda (la calculada hasta este punto durante todos los apartados anteriores) podemos hallar la solución a la ecuación diferencial sin restricciones. Una vez hecho esto, obtenemos un candidato último a posición y a velocidad, \mathbf{x}^{n+1} y \mathbf{v}^{n+1} , que será a los que les aplicaremos el método de la proyección múltiple para forzar que se cumpla la condición de inextensibilidad.

Lo primero que necesita este método es unos candidatos iniciales a velocidad y posición, \mathbf{v}_0 y \mathbf{x}_0 respectivamente, que han sido ya calculados como se indica anteriormente, así como el

vector $\|\mathbf{eR}\|$ con la medida de los puentes en el equilibrio. Tomamos un tamaño de paso para la resolución del método numérico, h , y la posición inicial como $X=X0 + hV0$, que iremos actualizando a lo largo del método. También requiere la matriz de masas de la cuerda M . Hay que recordar que si los extremos de la cuerda están fijos, tanto en el paso en el que se calculan los candidatos $X0$ y $V0$ como al calcular el valor X éstos han de mantenerse fijos.

Lo primero que hace este método es calcular un vector de restricciones C para los puentes de la cuerda que servirá para forzar la condición de inextensibilidad. Este vector de tamaño $n-1$ contendrá la diferencia de la norma al cuadrado de los candidatos a puentes (como los iniciales por ejemplo) entre la norma del puente en reposo menos la norma del puente en el reposo, es decir:

$$C_i = \frac{\|\mathbf{e}_i\|^2}{\|\mathbf{eR}_i\|} - \|\mathbf{eR}_i\|$$

Necesitamos también una variable umbral que servirá para medir el grado de error deseado en la hipótesis de inextensibilidad de la cuerda, es decir, si las restricciones se encuentran por debajo del umbral se considerará que la cuerda no se ha extendido (incluso puede usarse como un parámetro más de una cuerda extensible hasta cierto punto). A esta variable la llamaremos *umbral*.

Comenzamos ahora un bucle de resolución que se repita hasta que no se cumpla que todas las restricciones estén por debajo del umbral. Lo primero que hacemos es calcular el gradiente de las restricciones ∇C , dado por:

$$\nabla_{i+1} C_i = \frac{2\mathbf{e}_i}{\|\mathbf{eR}_i\|} ; \nabla_i C_i = -\frac{2\mathbf{e}_i}{\|\mathbf{eR}_i\|}$$

Los valores no contemplados son nulos. A esta matriz de tamaño $n-1 \times 3n$ que contiene al gradiente de las restricciones la llamaremos *grad_C*.

Con esta matriz, podemos calcular el vector $\delta\lambda$ al que llamaremos *d_lambda* que contendrá un incremento de los multiplicadores de Lagrange, dado por:

$$\delta\lambda = \frac{(\nabla C M^{-1} \nabla C^T)^{-1}}{h^2} C$$

Éste a su vez sirve para calcular un vector de un incremento de la posición δx al que llamaremos *d_x* que servirá para corregir las posiciones de los vértices para que la cuerda cumpla la hipótesis de inextensibilidad y que vendrá dado por

$$\delta \mathbf{x} = -h^2 M^{-1} \nabla C^T \delta \lambda$$

Lo siguiente es calcular la nueva posición de los vértices corregida, es decir, actualizar el vector de posiciones \mathbf{X} , de manera que

$$\mathbf{x} = \mathbf{x} + \delta \mathbf{x}$$

De nuevo hay que tener en cuenta las condiciones de contorno de la cuerda; si ésta esta fija ese incremento sólo debe aplicarse a los vértices no fijos.

Por último, actualizamos también el vector de restricciones \mathbf{C} , exactamente de la misma manera que lo hicimos anteriormente, pero con las nuevas posiciones de los vértices.

Aquí terminaría el bucle. En caso de no cumplirse las restricciones de la cuerda, se repetiría hasta que éstas se cumplieran. De cumplirse, las posiciones de los vértices corregidas que cumplen inextensibilidad vendrían dadas por el último vector de posiciones \mathbf{X} que se ha actualizado. La velocidad resultante de la cuerda vendría dada por

$$\mathbf{v} = \frac{(\mathbf{x} - \mathbf{x}_0)}{h}$$

Evolución temporal

Tenemos todas las herramientas necesarias para hallar la evolución temporal de nuestra cuerda. Lo primero que necesitamos ahora es dar un tiempo máximo para la simulación, al que llamaremos *time_max*, un tiempo inicial *time* con valor 0 y un paso temporal discreto al que llamaremos *t*. Esto conviene introducirlo al comienzo del programa para mayor comodidad.

Hecho esto, comenzamos el bucle de la simulación, donde se resolverán las ecuaciones del movimiento para cada instante de tiempo y acto seguido mediante VERLET se forzará inextensibilidad.

Primero actualizamos la variable temporal *time* sumándole el tiempo *t*, y luego resolvemos las ecuaciones para hallar $\mathbf{f}(\mathbf{x}^n)$. Ésta la obtenemos a partir de

$$\mathbf{f}(\mathbf{x}^n) = -\mathbf{M}^{-1} dE(\Gamma)/d\mathbf{x}$$

Una vez obtenida, resolvemos el sistema dado por

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}) \end{cases} \Rightarrow \begin{cases} \mathbf{v}^{n+1} = \mathbf{v}^n + t \mathbf{f}(\mathbf{x}^n) \\ \mathbf{x}^{n+1} = \mathbf{x}^n + t \mathbf{v}^{n+1} \end{cases}$$

Donde recordamos hay que tener en cuenta las condiciones de contorno de la cuerda. Este sistema de ecuaciones se puede modificar añadiendo por ejemplo una fuerza externa incluida en $f(x^n)$ (como la que definimos al principio del código). Ya tenemos nuestros candidatos X y V para la solución del problema. Forzamos ahora inextensibilidad con nuestra función VERLET, dando como argumentos de entrada la matriz de masas M la posición y la velocidad recién obtenidas X y V el paso t y las medidas de los puentes en reposo en el vector *medidas*. Obtenemos unos nuevos vectores velocidad y posición, V y X , que serán los corregidos. Para apreciar el desarrollo de la cuerda conviene dibujarla después de este paso.

Una vez halladas la nueva posición y velocidad, actualizamos los puentes y sus cantidades integradas en los vectores e y l . Ya sólo queda un último paso antes de repetir el bucle, y es comprobar que nuestro sistema Bishop del primer puente $b0$ no se haya desplazado a consecuencia del movimiento de la cuerda, violando así condiciones de perpendicularidad o no incluyendo al vector tangente del primer puente.

Lo que haremos será calcular el módulo del producto vectorial del primer vector del actual sistema Bishop $b0$ con el de la tangente en el primer puente obtenida después de la evolución temporal. Si este módulo es mayor que una tolerancia, significará que el primer puente se ha desplazado y por tanto habrá que rotar el sistema Bishop para que coincida con la configuración actual de la cuerda (tiene que contener a la tangente). Esta rotación es alrededor del vector producto vectorial del primer vector de $b0$ con el de la tangente en el primer puente.

Por último podemos actualizar las condiciones de contorno de la cuerda, por ejemplo añadirle giro en cada paso temporal.

Una vez hecho esto, volvemos al punto de **Construcción de los datos iniciales** y repetimos el proceso, construyendo los nuevos datos a partir de nuestro nuevo vector posición X y calculando las ecuaciones del movimiento para resolverlas y después aplicar VERLET.

Capítulo 3 – Código y ejemplos

En el capítulo anterior mostramos punto a punto cómo implementar un código a partir de la base teórica explicada en el capítulo 1, y ahora mostraremos un código ejemplo ya implementado junto con varias figuras de las simulaciones variando ciertos parámetros. Este código está hecho bajo la suposición de que ambos extremos de la cuerda están fijos, y se ha

retocado un poco con respecto al apartado anterior para acortarlo. También se incluirán las funciones auxiliares necesarias. El código está redactado en MATLAB.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ALGORITMO SIMULACION CUERDA DISCRETA TFG   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Introduccion de datos en reposo e iniciales   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Numero de vertices (n+2)
n = 25;                                %Mínimo 6 para que tenga sentido
t = 3e-3;                              %Tamaño de paso temporal
eje = [0 1.5 -0.15 0.15];              %Eje para la representacion grafica
eje2 = [-0.15 0.15 -0.15 0.15];        %Eje para la representacion grafica
eje3 = [0 1 0 1 -0.15 0.15];           %Eje para la representacion grafica
velsim = 1e-2;                         %Velocidad de simulación de la película

%Posicion de reposo
xR = linspace(0,1,n);
yR = linspace(0,1,n);
zR = linspace(0,0,n);

%Posicion inicial (extremos fijos)
xi = xR;
yi = yR;
zi = zR; zi(2)= 0.05;

%Reescritura de los datos
X = zeros(1,3*n);
for i=1:n
    X(3*i-2:3*i) = [xi(i),yi(i),zi(i)];
end
X = X';

%Velocidad inicial
V = zeros(1,3*n);
V = V';

%Vector con la fuerza externa ejercida sobre la cuerda
F_ext = zeros(3*n,1);

%Parametros
alpha = 0.1;                          %rod's bending modulus
beta = 10;                             %rod's twisting modulus
M = eye(3*n)*50;                      %Matriz de masas de los vertices

```

```

B = alpha*eye(2);           %Matriz de flexion (uniforme)

%Puentes y cantidades integradas en el equilibrio
eR = zeros(n-1,3);
for i=1:n-1
    eR(i,:) = [xR(i+1),yR(i+1),zR(i+1)]-[xR(i),yR(i),zR(i)];
end
lR = zeros(1,n-2);
for i=1:n-2
    lR(i) = norm(eR(i,:))+norm(eR(i+1,:));
end

%Puentes y cantidades integradas
e = zeros(n-1,3);
for i=1:n-1
    e(i,:) = [xi(i+1),yi(i+1),zi(i+1)]-[xi(i),yi(i),zi(i)];
end
l = zeros(1,n-2);
for i=1:n-2
    l(i) = norm(e(i,:))+norm(e(i+1,:));
end

%Bishop frame en el primer puente.
b0 = [eR(1,:);[1,-1,0];cross(eR(1,:),[1,-1,0])]; % [t,u,v]
if (b0(1,:)*b0(2,:))' ~= 0
    disp('Datos erroneos en b0')
    return
end

for i=1:3
    b0(i,:) = b0(i,:)/norm(b0(i,:));
end

%Condiciones de contorno
theta = zeros(1,n-1);
theta(1) = 3*pi;
theta(n-1) = 0;
K = (theta(n-1)-theta(1))/sum(lR);
%Incremento temporal de theta
d_thetai = 0;
d_thetaf = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Construccion de datos en reposo   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Curvatura binormal en reposo
kbR = zeros(n-2,3);
for i=1:n-2
    kbR(i,:) = 2*cross(eR(i,:),eR(i+1,:))/(norm(eR(i,:))*norm(eR(i+1,:)) +
eR(i,:)*eR(i+1,:))';
end

%Tanto el Bishop como el material frame son ternas de vectores asociadas a
%cada uno de los puentes de la cuerda. Se guardaran a partir de una

```

```

%sucesion de matrices, o de una matriz de 3x3xn-1, donde la matriz i-esima
%sera la asociada al puente i-esimo, el primer vector sera el tangente, el
%segundo el vector u o m1 y el tercero el v o m2

%Bishop frame en reposo y transporte paralelo
bishopR = zeros(3,3,n-1);
bishopR(:, :, 1) = b0;
for i=2:n-1
    %Cuerda recta en el equilibrio
    if sum(kbR(i-1,:)==[0 0 0]) == 3
        bishopR(1,:,i) = bishopR(1,:,i-1);
        bishopR(2,:,i) = bishopR(2,:,i-1);
        bishopR(3,:,i) = bishopR(3,:,i-1);
    %Cuerda no recta en el equilibrio
    else
        [~,matriz] = rotacion(eR(i-1,:),eR(i,:),kbR(i-1,:));
        bishopR(1,:,i) = (matriz*bishopR(1,:,i-1))';
        bishopR(2,:,i) = (matriz*bishopR(2,:,i-1))';
        bishopR(3,:,i) = cross(bishopR(1,:,i),bishopR(2,:,i));
    end
end

%Giro entre Bishop y material frame en reposo
%Se empieza con el bishop y se rota hasta el material
thetaR = zeros(1,n-1);

%Calculo del material frame en reposo
mframeR = zeros(3,3,n-1);
for i=1:n-1
    mframeR(1,:,i) = bishopR(1,:,i);
    mframeR(2,:,i) = cos(thetaR(i))*bishopR(2,:,i) +
sin(thetaR(i))*bishopR(3,:,i);
    mframeR(3,:,i) = -sin(thetaR(i))*bishopR(2,:,i) +
cos(thetaR(i))*bishopR(3,:,i);
end

%Curvatura material en reposo
wR = zeros(n-2,2,2);
wR(1, :, :) = [kbR(1,:) * mframeR(3, :, 1)', -
kbR(1,:) * mframeR(2, :, 1)'; kbR(1,:) * mframeR(3, :, 2)', -
kbR(1,:) * mframeR(2, :, 2)']';
for i=2:n-2
    wR(i, :, :) = [kbR(i,:) * mframeR(3, :, i)', -
kbR(i,:) * mframeR(2, :, i)'; kbR(i,:) * mframeR(3, :, i+1)', -
kbR(i,:) * mframeR(2, :, i+1)']';
end

%Variable de tiempo de simulacion
time = 0;
time_max = 0.5;

%Cuerda inicial
subplot(2,2,1)
plot(sqrt(2)*X(1:3:end),X(3:3:end))
axis(eje)

```

```

subplot(2,2,2)
plot(sqrt(2)/2*(X(1:3:end)-X(2:3:end)),X(3:3:end))
axis(eje2)
subplot(2,2,3)
plot3(X(1:3:end),X(2:3:end),X(3:3:end));
axis(eje3)
pause(velsim)

%Medidas de los puentes en reposo
medidas = zeros(1,n-1);
for i=1:n-1
    medidas(1,i) = norm(eR(i,:));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   EVOLUCION TEMPORAL   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Definimos tamaños para mejorar velocidad del codigo
kb = zeros(n-2,3);
mframe = zeros(3,3,n-1);
w = zeros(n-2,2,2);
grad_kb = zeros(3,3,n,n-2);
grad_phi = zeros(3,n,n-2);
grad_angrot = zeros(3,n,n-2);
grad_w = zeros(2,3,n,n-2,2);

while time<time_max
    time = time + t;

    %Curvatura binormal
    for i=1:n-2
        kb(i,:) = 2*cross(e(i,:),e(i+1,:))/(norm(eR(i,:))*norm(eR(i+1,:)) +
e(i,:)*e(i+1,:)');
    end

    %Bishop frame
    bishop = zeros(3,3,n-1);
    bishop(:, :, 1) = b0;
    for i=2:n-1
        %Cuerda recta
        if sum(kb(i-1,:)==[0 0 0]) == 3
            bishop(1,:,i) = bishop(1,:,i-1);
            bishop(2,:,i) = bishop(2,:,i-1);
            bishop(3,:,i) = bishop(3,:,i-1);
        %Cuerda no recta
        else
            [~,matriz] = rotacion(e(i-1,:),e(i,:),kb(i-1,:));
            bishop(1,:,i) = (matriz*bishop(1,:,i-1))';
            bishop(2,:,i) = (matriz*bishop(2,:,i-1))';
            bishop(3,:,i) = cross(bishop(1,:,i),bishop(2,:,i));
        end
    end
end

%Calculo del giro entre Bishop y material frame

```

```

for i=2:n-2
    theta(i) = K*lR(i) + theta(i-1);
end

%Material frame
for i=1:n-1
    mframe(1,:,i) = bishop(1,:,i);
    mframe(2,:,i) = cos(theta(i))*bishop(2,:,i) +
sin(theta(i))*bishop(3,:,i);
    mframe(3,:,i) = -sin(theta(i))*bishop(2,:,i) +
cos(theta(i))*bishop(3,:,i);
end

%Curvatura material inicial
w(1, :, :) = [kb(1, :)*mframe(3, :, 1)', -
kb(1, :)*mframe(2, :, 1)'; kb(1, :)*mframe(3, :, 2)', -kb(1, :)*mframe(2, :, 2)']';
for i=2:n-2
    w(i, :, :) = [kb(i, :)*mframe(3, :, i)', -
kb(i, :)*mframe(2, :, i)'; kb(i, :)*mframe(3, :, i+1)', -kb(i, :)*mframe(2, :, i+1)']';
end

%Gradiente de la curvatura binormal
for i=1:n-2
    grad_kb(:, :, i, i) = (2*matriz_vectorial(e(i+1, :)) +
kb(i, :)*e(i+1, :))/(norm(eR(i, :))*norm(eR(i+1, :)) + e(i, :)*e(i+1, :));
    grad_kb(:, :, i+2, i) = (2*matriz_vectorial(e(i, :)) -
kb(i, :)*e(i, :))/(norm(eR(i, :))*norm(eR(i+1, :)) + e(i, :)*e(i+1, :));
    grad_kb(:, :, i+1, i) = -(grad_kb(:, :, i, i) + grad_kb(:, :, i+2, i));
end

%Gradiente de holonomia de conexion
for i=1:n-2
    grad_phi(:, i, i) = kb(i, :)/(2*norm(eR(i, :)));
    grad_phi(:, i+2, i) = -kb(i, :)/(2*norm(eR(i+1, :)));
    grad_phi(:, i+1, i) = -( grad_phi(:, i, i) + grad_phi(:, i+2, i));
end

%Gradiente del angulo resultante de giro
for i=1:n
    for j=1:n-2
        grad_angrot(:, i, j+1) = sum(grad_phi(:, i, 1:j), 3);
    end
end

%Rotacion pi/2 en contra de las agujas del reloj
J = [0 1; -1 0];
for i=1:n
    for k=1:n-2
        cont = k; %Contador auxiliar para las cuentas
        for j=1:2
            grad_w(:, :, i, k, j) = [mframe(3, :, cont); -
mframe(2, :, cont)]*grad_kb(:, :, i, k) -

```

```

J*[w(k,j,1);w(k,j,2)]*[grad_angrot(1,i,cont),grad_angrot(2,i,cont),grad_angro
t(3,i,cont)];
    cont = cont+1;
end
end
end

%Calculo energias/fuerzas en los vertices
dE_theta = 1/lR(n-2)*([w(n-2,2,1),w(n-2,2,2)]*J*B*[w(n-2,2,1)-wR(n-
2,2,1);w(n-2,2,2)-wR(n-2,2,2)] + 2*beta*(theta(n-1)-theta(1)));
dE_x = zeros(1,3*n);
aux = zeros(3,1);
for i=1:n %dE/dx(i)
    for k=1:n-2
        for j=1:2
            aux(:,1) = aux(:,1) + grad_w(:, :, i, k, j)'*B*[w(k,j,1)-
wR(k,j,1);w(k,j,2)-wR(k,j,2)];
        end
        dE_x(1,3*i-2:3*i) = dE_x(1,3*i-2:3*i) + aux(:,1)'/lR(k);
        aux = zeros(3,1);
    end
end

%Derivada del ultimo theta con respecto a x(i)
dphi = zeros(1,3*n);
%Contemplamos los 4 únicos casos que no tienen tres sumandos
dphi(1:3) = grad_phi(:,1,1)';
dphi(4:6) = grad_phi(:,2,1)' + grad_phi(:,2,2)';
dphi(3*(n-1)-2:3*(n-1)) = grad_phi(:,n-1,n-3)' + grad_phi(:,n-1,n-2)';
dphi(3*n-2:3*n) = grad_phi(:,n,n-2)';
for i=3:n-3
    dphi(1,3*i-2:3*i) = grad_phi(:,i,i-2)' + grad_phi(:,i,i-1)' +
grad_phi(:,i,i)'; %Esto esta casi antioptimizado...
end

%Derivada total de la energia
dE = dE_x - dE_theta*dphi;

%Resolucion sin restriccion inextensibilidad
f = -inv(M)*dE' + F_ext;
V(4:3*(n-1)) = V(4:3*(n-1)) + t*f(4:3*(n-1));
X(4:3*(n-1)) = X(4:3*(n-1)) + t*V(4:3*(n-1));

%Forzamos inextensibilidad
[X,V] = VERLET(M,X,t,medidas,V);

%Dibujo de la cuerda
subplot(2,2,1)
plot(sqrt(2)*X(1:3:end),X(3:3:end))
axis(eje)
subplot(2,2,2)
plot(sqrt(2)/2*(X(1:3:end)-X(2:3:end)),X(3:3:end))
axis(eje2)
subplot(2,2,3)
plot3(X(1:3:end),X(2:3:end),X(3:3:end));

```

```

axis(eje3)
pause(velsim)

%Puentes y cantidades integradas
for i=1:n-1
    e(i,:) = X(3*i+1:3*(i+1))' - X(3*i-2:3*i)';
end

for i=1:n-2
    l(i) = norm(e(i,:))+norm(e(i+1,:));
end

%Actualizacion del sistema Bishop
kb0 = cross(b0(1,:),e(1,+)/norm(e(i,:)));
if norm(kb0) > 1e-3
    [~,matriz] = rotacion(b0(1,:),e(1,:),kb0);
    b0(1,:) = (matriz*b0(1,:))';
    b0(2,:) = (matriz*b0(2,:))';
    b0(3,:) = (matriz*b0(3,:))';
end

%Actualizacion de condiciones de contorno
theta(1) = theta(1) + d_thetai;
theta(n-1) = theta(n-1) + d_thetaf;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo de la proyeccion multiple %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Funcion que fuerza inextensibilidad para solucion de x''=f(x)
%Usa el metodo de la proyeccion multiple
%M = matriz de masas tamaño 3n*3n
%x0,v0 condiciones iniciales (vectores 3n con la posicion y velocidad)
%h: vector con tamaño de paso
%eR: medida de los puentes en reposo. Sirve para calcular C
function [x,v] = VERLET(M,x0,h,eR,v0)

n = length(x0)/3;

%Candidatos a puntos
x = x0; v = v0;
%Posicion inicial
x(4:3*(n-1)) = x0(4:3*(n-1)) + v0(4:3*(n-1))*h;

%Restricciones C(x), grad_C(x)
C = zeros(n-1,1);
for i=1:n-1
    C(i) = (norm(x0(3*i+1:3*(i+1)) - x0(3*(i-1)+1:3*i)))^2/eR(i) - eR(i);
end
grad_C = zeros(n-1,3*n);

```



```

%Tolerancia de extensibilidad
umbral = 1e-2;
while max(abs(C))>umbral

    %Calculo de las restricciones
    for i=1:n-1
        aux = x(3*i+1:3*(i+1)) - x(3*i-2:3*i);
        grad_C(i,3*i-2:3*i) = -2*aux/eR(i);
        grad_C(i,3*i+1:3*i+3) = -grad_C(i,3*i-2:3*i);
    end

    %Multiplicador Lagrange
    d_landa = ((grad_C*(M\grad_C'))\C)/h^2;

    %Variacion de la posicion
    d_x = -h^2*(M\grad_C')*d_landa;

    %Correccion de la posicion
    x(4:3*(n-1)) = x(4:3*(n-1)) + d_x(4:3*(n-1));

    %Calculo del umbral
    for i=1:n-1
        C(i) = (norm(x(3*i+1:3*(i+1)) - x(3*(i-1)+1:3*i)))^2/eR(i) - eR(i);
    end

end

%Velocidad corregida
v(4:3*(n-1)) = (x(4:3*(n-1)) - x0(4:3*(n-1)))/h;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Matriz de rotacion   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Funcion que calcula la matriz de rotacion de angulo phi entorno a un eje
%arbitrario mediante el uso de cuaterniones
%El giro es antihorario
function [matriz] = matriz_rotacion(phi,v)

%Cuaternion y matriz que definen el giro
q = [cos(phi/2), sin(phi/2)*v/norm(v)];
matriz = [q(1)^2+q(2)^2-q(3)^2-q(4)^2, 2*(q(2)*q(3)-
q(1)*q(4)), 2*(q(1)*q(3)+q(2)*q(4)), 2*(q(2)*q(3)+q(1)*q(4)), q(1)^2-
q(2)^2+q(3)^2-q(4)^2, 2*(q(3)*q(4)-q(1)*q(2)), 2*(q(2)*q(4)-
q(1)*q(3)), 2*(q(1)*q(2)+q(3)*q(4)), q(1)^2-q(2)^2-q(3)^2+q(4)^2]/norm(q)^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Funcion rotacion   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Funcion que calculara a partir del uso de cuaterniones la rotacion de los
%sistemas de referencia entorno al eje deseado.

```

```

%e0 y e1 son los puentes entre los que se quiere calcular la rotacion
%v es el eje de giro
%Usa la funcion externa de la matriz de rotacion
%Los vectores e0,e1 y v son de 1xn

function [phi,matriz] = rotacion(e0,e1,v)

%Calculo del angulo de giro entre e0 y e1
phi = acos(e0*e1'/(norm(e0)*norm(e1)));
%Corregimos posibles errores numericos
phi = real(phi);

%Comprobamos el sentido de giro
aux0 = (matriz_rotacion(phi,v)*e0'/norm(e0))';
aux1 = e1/norm(e1);
tolerancia = 1e-6;

%Si el resultado de rotar e0 no es el vector e1
if sum(abs(aux0-aux1)<tolerancia) ~= 3
    phi = -phi;
end

%Matriz que define el giro
matriz = matriz_rotacion(phi,v);

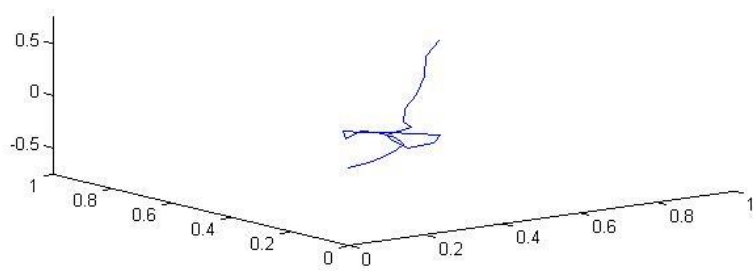
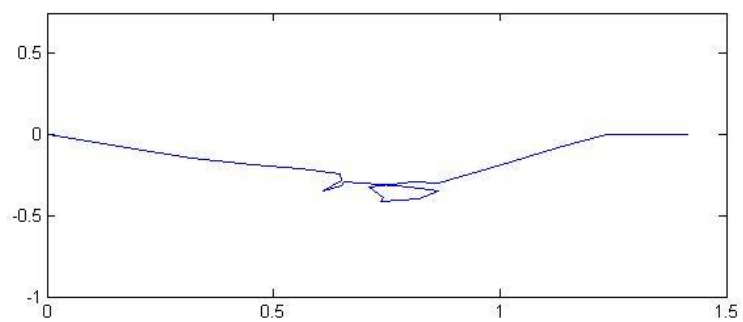
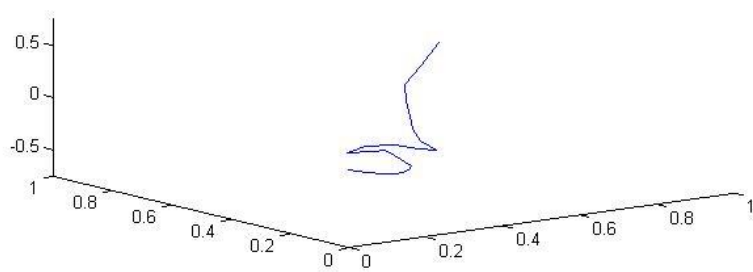
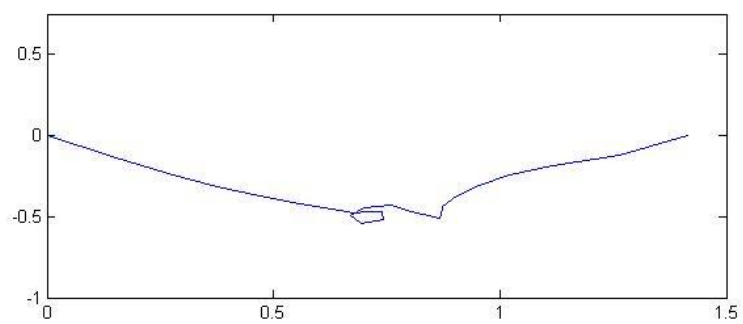
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Funcion matriz producto vectorial   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Funcion que dado un vector "e" de dimension 3 encuentra la matriz E que
cumple que
%cross(e,x) = Ex, para todo vector x.
%El vector e ha de introducirse como vector fila

function [E] = matriz_vectorial(e)
E = [0 -e(3) e(2); e(3) 0 -e(1); -e(2) e(1) 0];

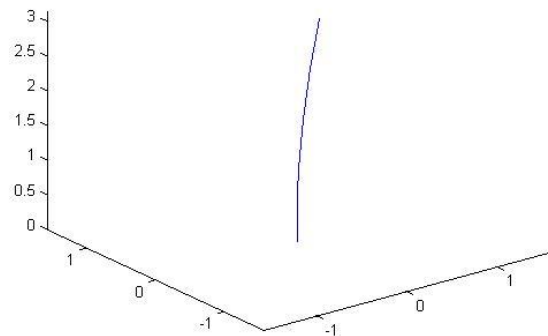
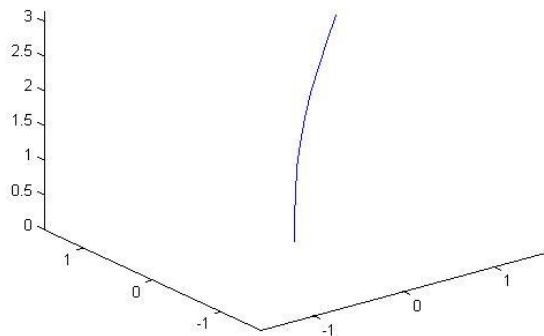
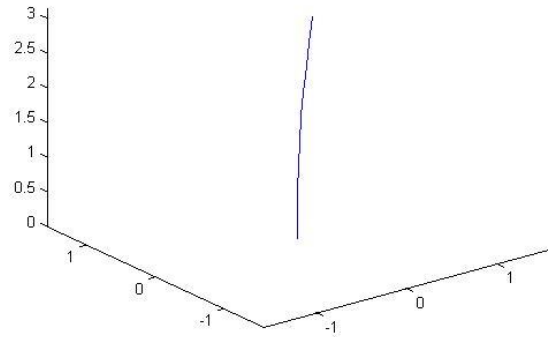
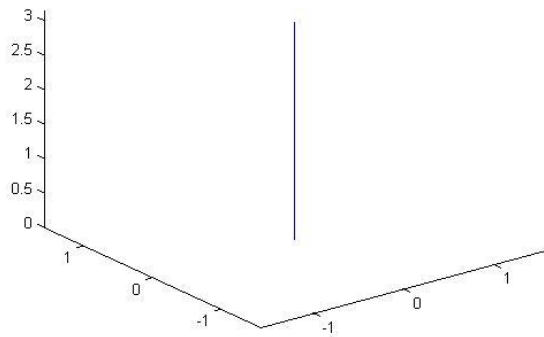
```

Éste es el código realizado para una cuerda recta en el equilibrio a la que se perturba ligeramente cerca de uno de los extremos en el que además se aplica un giro de 3π . Retocando ciertos parámetros del código podemos obtener otras simulaciones también muy interesantes, como la siguiente en la que la posición inicial y de equilibrio de la cuerda es una catenaria con ecuación $z(x) = \cosh(4.35464x - 2.17732)/4.35464 - 1.02602$; $y = x$, pero a la que se le aplica un giro de ángulo π en cada uno de los extremos y con sentido opuesto. Obtenemos algunas imágenes como estas:



Se puede observar cómo en este caso la cuerda, aunque inextensible, tiene libertad de movimiento, y por tanto se forman espirales y se dobla sobre sí misma tal y como pasaría en un caso real.

Un último ejemplo es el de una cuerda con un extremo libre. En este ejemplo se dará una cuerda rígida, mucho más similar a una varilla, subiendo el parámetro de flexión α . Intentaremos simular de manera sencilla el movimiento de un árbol que soporta una fuerza externa como podría ser un viento constante, y lo haremos con una varilla que estará sujeta al suelo y con el extremo superior libre. En este caso como un extremo está libre no hay energía de torsión, con lo que todo el movimiento será debido a la energía de flexión.



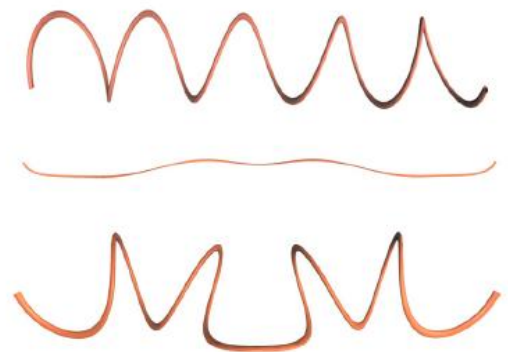
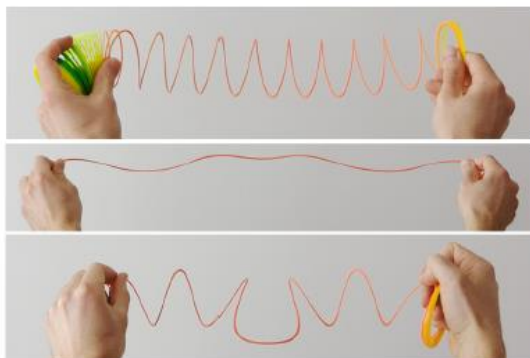
Se puede observar cómo nuestra varilla se comba hasta un cierto punto a causa del viento y luego recupera su posición anterior, siguiendo un movimiento oscilatorio, de una manera similar a un caso real. Si se hiciera un estudio previo del problema y variáramos los parámetros de manera conveniente podríamos simular situaciones más complejas y más cercanas a la realidad.

Conclusión

Este trabajo ha mostrado cómo con un sencillo método se pueden modelizar situaciones complejas de la vida real para posteriormente simularlas con una gran precisión. Se han usado conocimientos de geometría diferencial estrechamente ligados a la física del problema, y se ha hecho uso de métodos numéricos eficientes para resolverlo. Los ejemplos mostrados son sólo un pequeño ejemplo de todo lo que este método puede dar de sí, pudiendo ahondar en este campo y conseguir simular figuras como las siguientes obtenidas del artículo “Discrete elastic rods” (ver referencias), donde se comprueba el asombroso grado de precisión que tiene este método:



Estas imágenes corresponden a la modelización del comportamiento de un nudo sometido a rotaciones de los extremos de la cuerda, y usando este método predecimos casi sin error su comportamiento. Otro caso interesante es el de la deformación de un muelle por estiración, como muestra la siguiente figura:



Se observa que el modelo predice y simula el comportamiento del muelle después de sufrir una estiración crítica con un grado de precisión espectacular.

También se puede añadir componentes rígidas al método, y simular situaciones aparentemente distintas al caso de las cuerdas, como es el caso de la siguiente figura en la que se simula un árbol bajo la acción del viento:



Vemos así que este método no queda reducido al estudio de las cuerdas sino que también abarca situaciones muy distintas que a priori no estarían relacionadas entre sí y cuya modelización y resolución podría ser complicada. Todo esto convierte este método en una forma muy interesante de simular comportamientos reales e incluso de entender cómo funciona la realidad física que observamos.

Referencias

- Bergou, M.; Wardetzky, M.; Robinson, S.; Audoly, B.; Grinspun, E. Discrete elastic rods. ACM Transactions on Graphics (SIGGRAPH), vol. 27, n 3 (2008), p 63:1 - 63:12
- Goldenthal, R.; Harmon, D.; Fattal, R.; Bercovier, M.; Grinspun, E.; Efficient Simulation of Inextensible Cloth. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) , vol. 26, n3 (2007)
- Espeso D.R., Carpio A., Martinez-Garcia E., de Lorenzo V., Stenosis triggers spread of helical Pseudomonas biofilms in cylindrical flow systems, Scientific Reports 6 (2016) 27170.
- Valdés, A. Notas de Geometra Diferencial con aplicaciones. Geometría diferencial y sus aplicaciones. Universidad Complutense de Madrid. 22 de enero de 2013
- Bergou, M.; Wardetzky, M.; Robinson, S.; Audoly, B.; Grinspun, E. Geometric aspects of discrete elastic rods. Oberwolfach Reports, vol. 6, primera edicin, 2009
- Ambrose, W.; Singer, I.M. A theorem on holonomy. Transactions of the American Mathematical Society, vol. 75, n 3 (1953), p 428-443